Strategies for Debugging Parallel Applications

Session 643

These are confidential sessions—please refrain from filming or taking pictures.

Enterprise IT

Strategies for Debugging Parallel Applications

Steve Peters, PhD
Mac OS Platform Performance Engineering

http://developer.apple.com/wwdc2005/

Session Overview

- A personal perspective on cluster “debugging” (mine)
- Shark—an introduction to old and new features
- Absoft FxP Technology
- Etnus TotalView on Mac OS X

Our Involvement With Cluster Computing

- Assisted with bring–up and “Top 500” benchmarks for Virginia Tech’s System X. Both the G5 PowerMac and G5 XServe configurations
- Helped tune commissioning benchmarks for COLSA’s Mach-5 XServe cluster
- Meet with HPC developers one-on-one in performance “kitchens”
- Consult internally on RFPs

Roadmap for This Talk

- For developing *new* HPC code on Mac OS X
  - A rundown of implementation, performance, and debugging aids

- When bringing *existing* HPC code to Mac OS X…
  - Debugging == "Performance Improvement" (90% of the time)
  - A survey of Mac OS X performance "high ground"
  - Rules of thumb and some cautions for bring-up

Implementing a New Code

- Consider Apple's tested and tuned libraries
  - BLAS, LAPACK
  - vMathlib—single precision, 4 at a time, elementary math functions
  - vForce—single and double, array of memory operands, math functions
  - vDSP—single and double Digital Signal Processing, with very fast FFTs
  - vImage—image processing, many pixel formats
- Consider auto-vectorization tools, e.g. Absoft VAST, gcc 4.0
- Follow a cautious bring-up discipline (like the one we'll soon describe)
- Consider TotalView® for parallel debugging
- Tune with Shark

Everything Old Is New Again—Ported Codes

- Existing code—Already builds and runs on a cluster—Port to Mac OS X
  - Often "it just works," but leaves performance on the table
- Config or Makefile issues
  - Aware of the platform? E.g. autconfig scripts that grok "Darwin"
  - Tuned for the platform? E.g. machine dependent implementations
- Compiler issues
  - Using performance compiler?
  - Using performance options?
- Library issues
- Which MPI?
- Watch out for fallback "reference implementations" of e.g. BLAS, FFTs (usually selected by some #ifdef)

A Bring-up Plan

- Start small and work the size-up
- Verify fabric, communication performance with vendor tools
- Expect issues as power-of-two boundaries are crossed
- Expect issues as physical boundaries are crossed (rack, switch, aisle, …)
- Investigate all scaling anomalies
- Do A/B comparisons on symmetric subsets of the cluster—Start small

Special Topic: HPC-Friendly Memory Allocation

- Mac OS X userland memory allocator (malloc) is tuned for desktop
  - Favors small, short lived allocations
  - Aggressively releases large allocations to minimize memory pressure
  - No preferred access pattern
- HPC may prefer
  - Large, long lived allocations (i.e., big data arrays)
  - Sequential or fixed stride access
  - Efficient use of memory cache hierarchy is essential!
- Want to ensure contiguous virtual address range, e.g. an array, is backed by contiguous physical pages
  - Sequential access gets full benefit of L2 cache
  - Goto's BLAS requires this scheme for best performance
  - Communication layer enjoys benefits in message passing (wired pages)

Special Topic: HPC-Friendly Memory Allocation

- Sample code: HPCMemory—contact: skip@apple.com
  - Kernel Extension (.kext) grabs a contiguous extent of physical memory
  - Library interfaces permit connection to kext, allocation, and free

```
extern kern_return_t HPCMemOpen(HPCMemRef *memRefPtr);
extern void HPCMemClose(HPCMemRef memRef);

extern kern_return_t HPCMemAllocate(HPCMemRef memRef, size_t size, void **allocPtr);
extern void HPCMemFree(HPCMemRef memRef, void *addr);

extern kern_return_t HPCMemCheckContig(HPCMemRef memRef, const void *addr, size_t size,
bool *isContig);
```

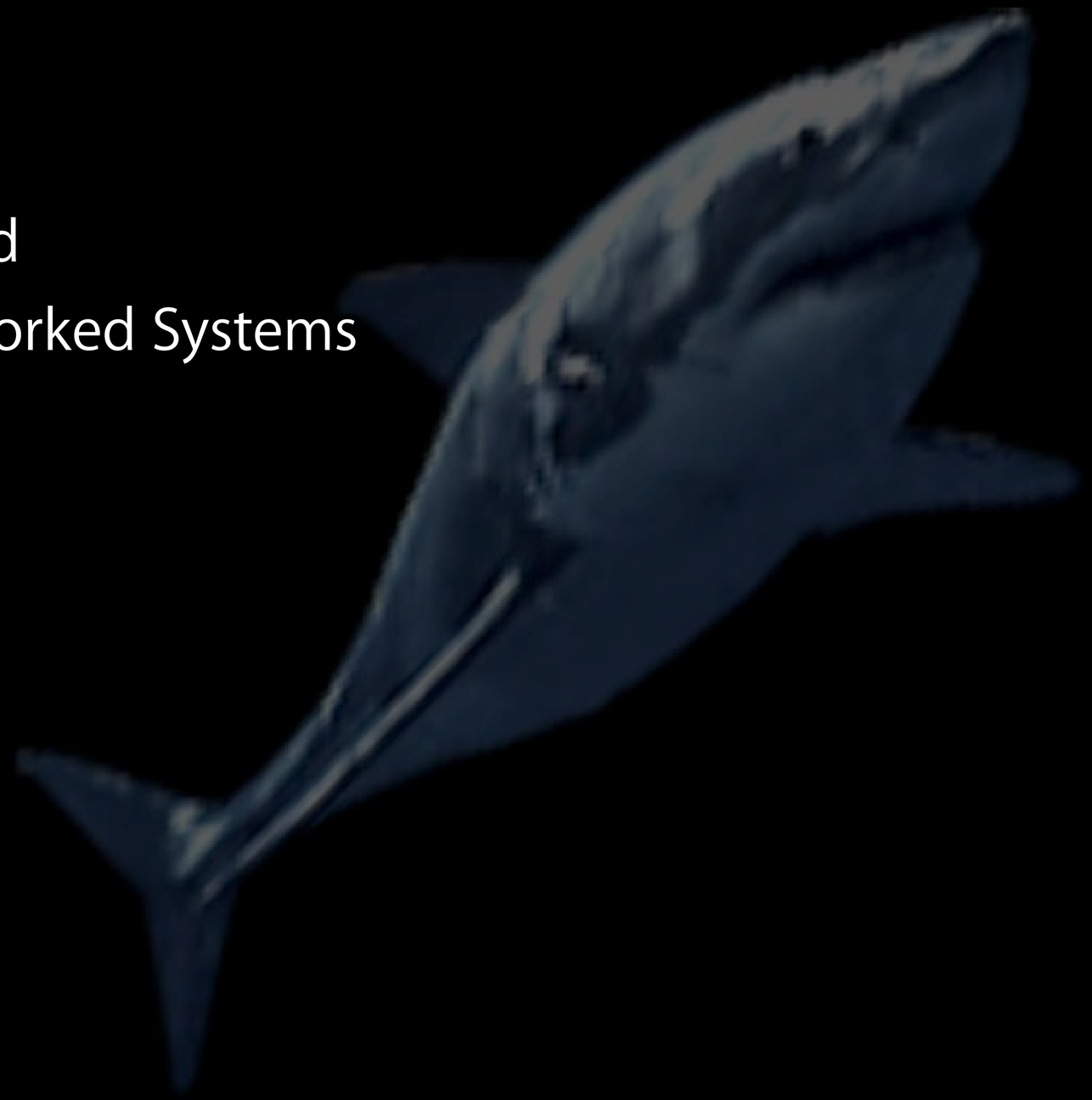
Profiling Networked Systems With Shark

Yusuf Abdulghani
Performance Architect

http://developer.apple.com/wwdc2005/

Overview

- What is Shark?
- Getting Started
- Profiling Networked Systems

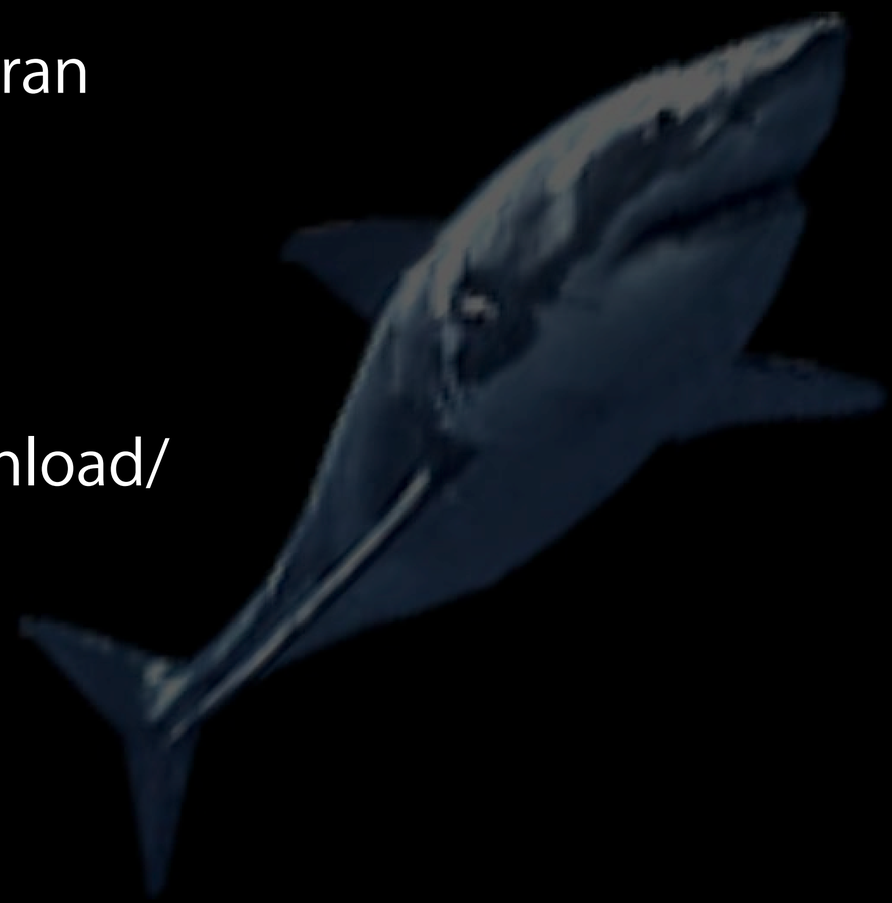


What Is Shark?

http://developer.apple.com/wwdc2005/

Shark

- Simple and fast profiling tool for Mac OS X
- Runs on 10.3 (Panther) and 10.4 (Tiger)
- Works with the language and compiler of your choice
  - C, C++, ObjC, Java, Fortran
  - GCC, CodeWarrior, XLC/XLF, Absoft Fortran
  - Mach-O, CFM
- GUI or command-line
- CHUD 4.2 Preview available today
  - http://developer.apple.com/tools/download/
- **Free!**



Shark Workflows

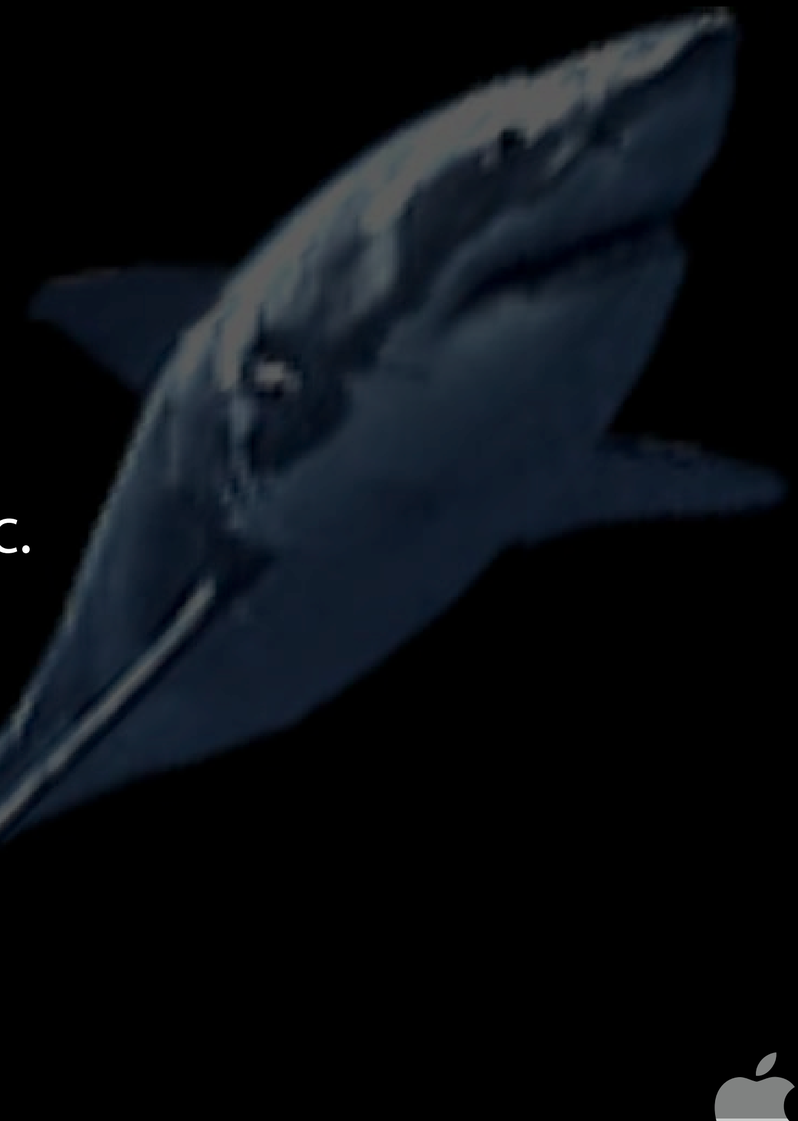
Many paths to enlightenment

- Time profiling
- Malloc tracing
- Java profiling
- Counter spreadsheet
- Custom configuration
- Network profiling

Time Profile

A brief history of your computer

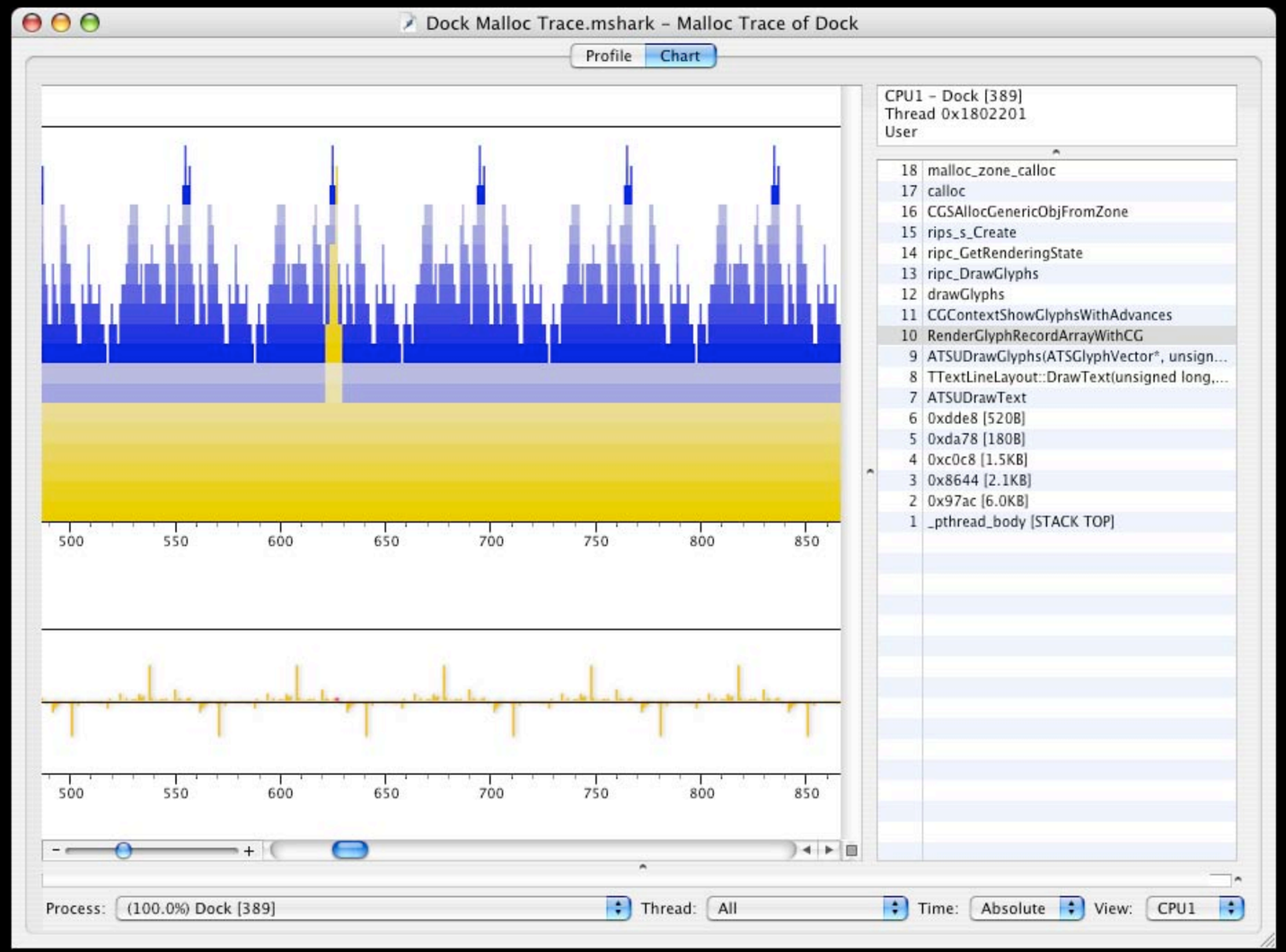
- The most common workflow
  - Shows you where time is being spent
  - Optimize performance-critical code
- Statistical sampling
  - 1,000 data points every second (default)
  - Each sample contains backtrace, process, thread, etc.
- Very low overhead
- Captures **everything**—drivers, kernel, applications



Malloc Tracing

Thin is in

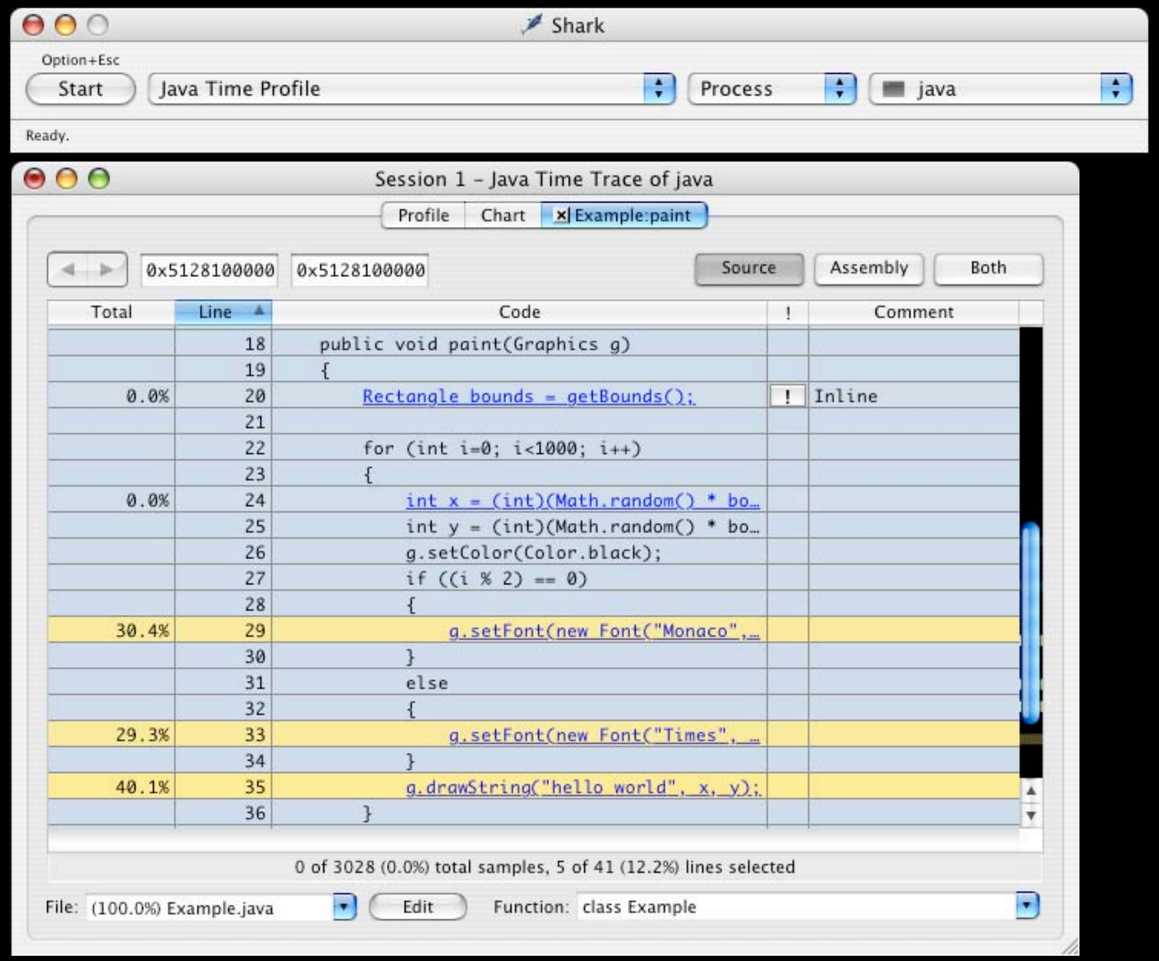
- Monitor allocations (`malloc`, `new`, etc.)
- Understand application memory usage
- Visualize program behavior



Java Profiling

Brew the perfect cup

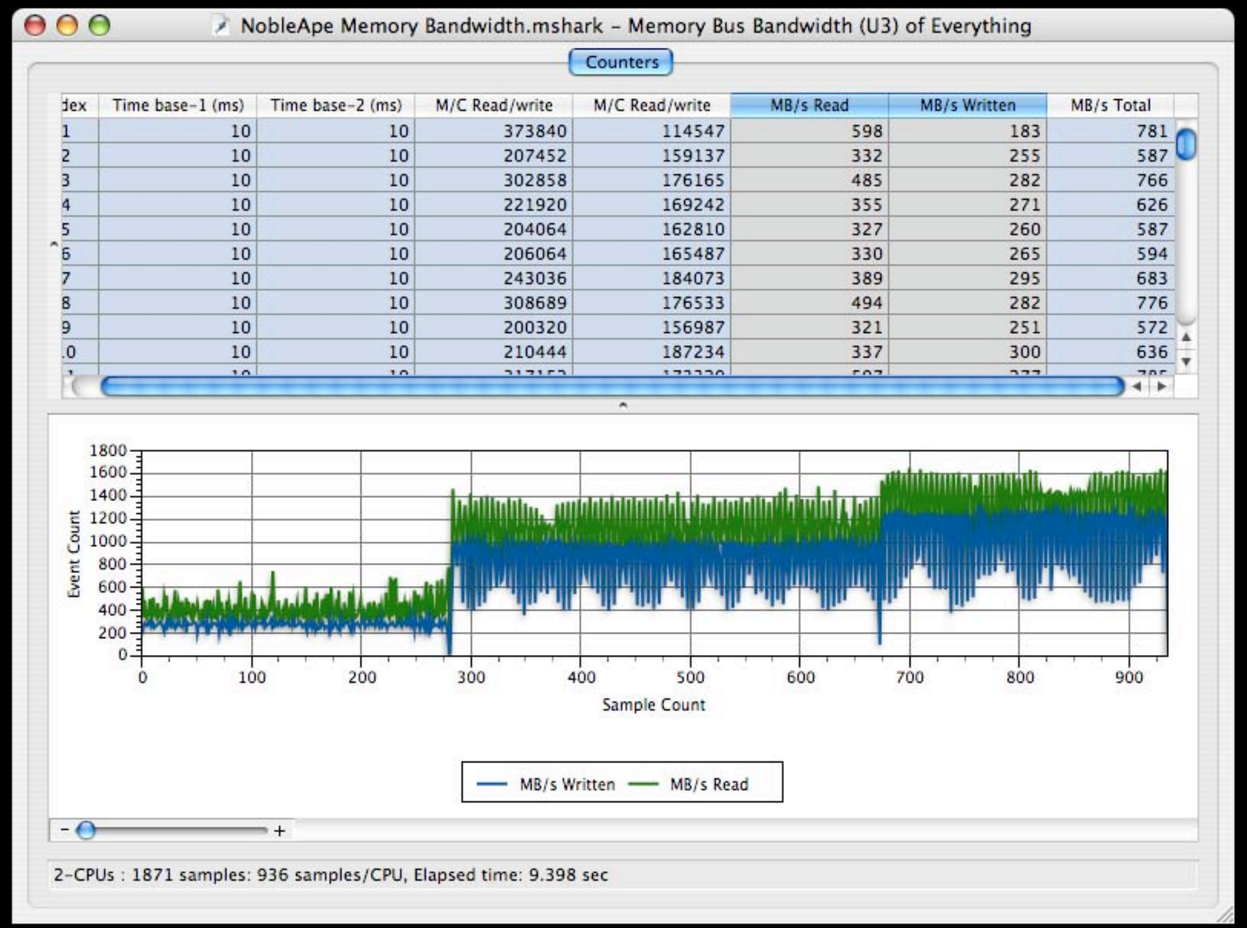
- Three configs
  - Time profiling
  - Alloc tracing
  - Function tracing
- View source in Code Browser



Counter Spreadsheet

Add it all up

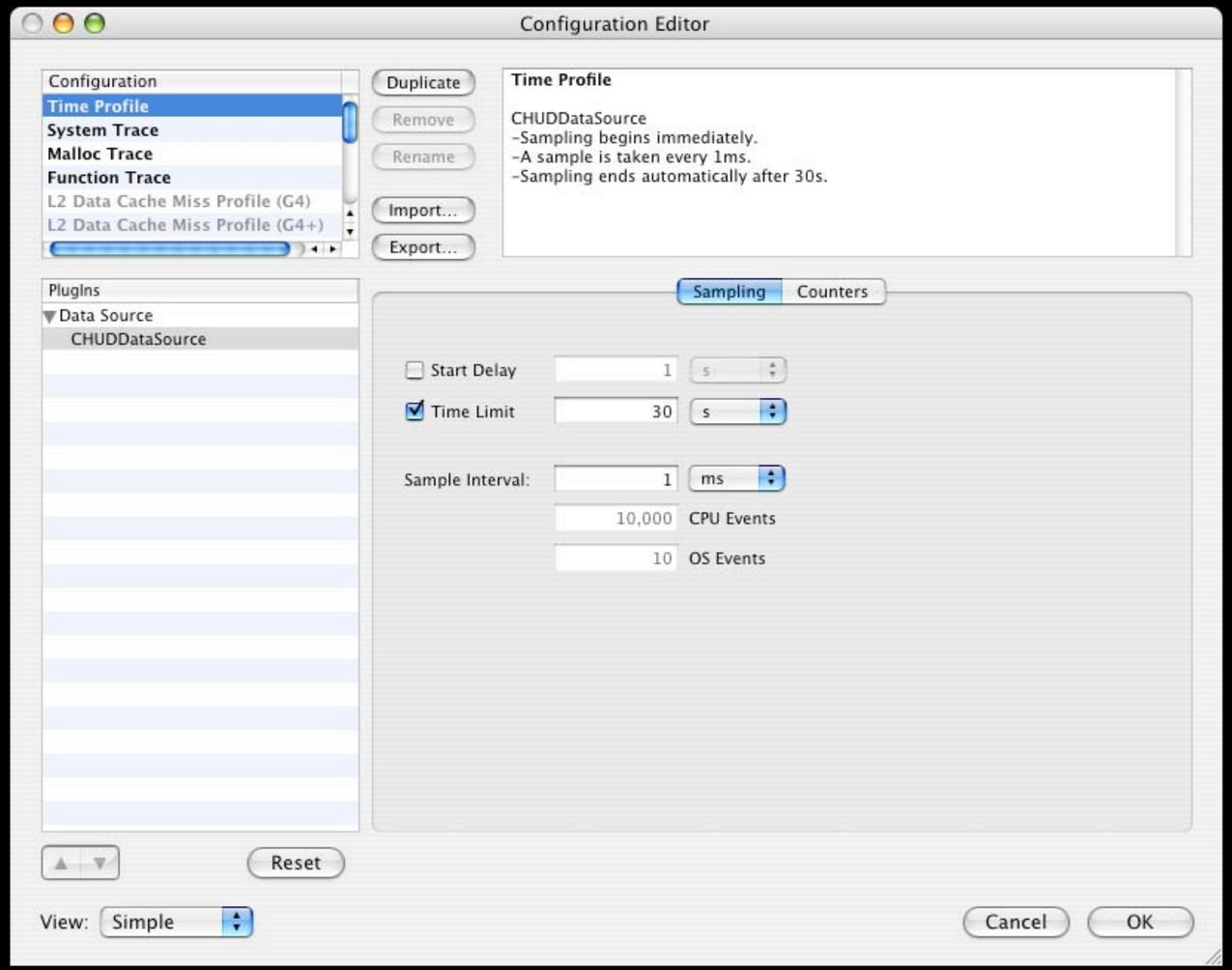
- Quantify performance events (memory bandwidth, CPI)
- Design your own metrics



Configuration Editor

The buck stops here

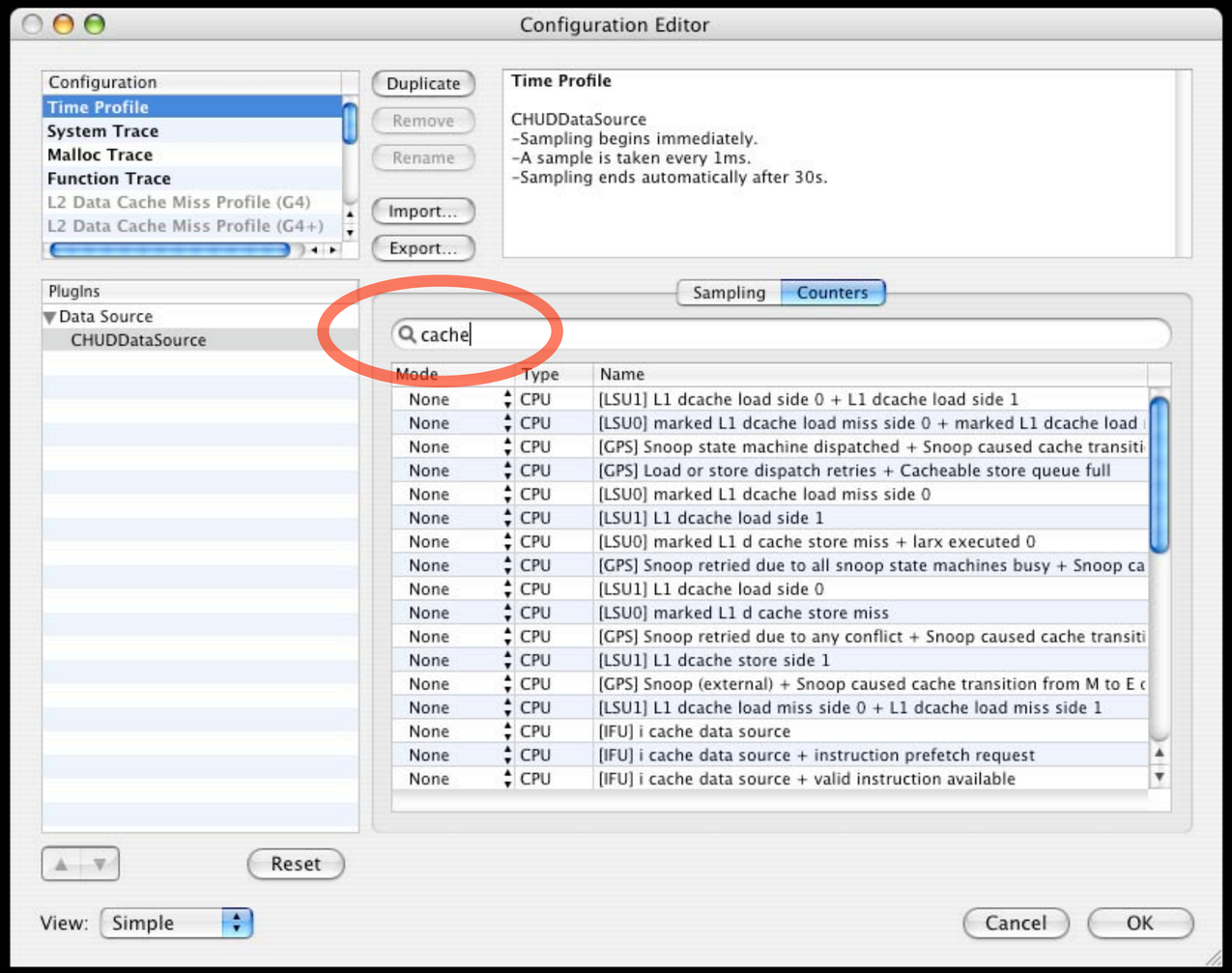
- Create your own configs



Configuration Editor

The buck stops here

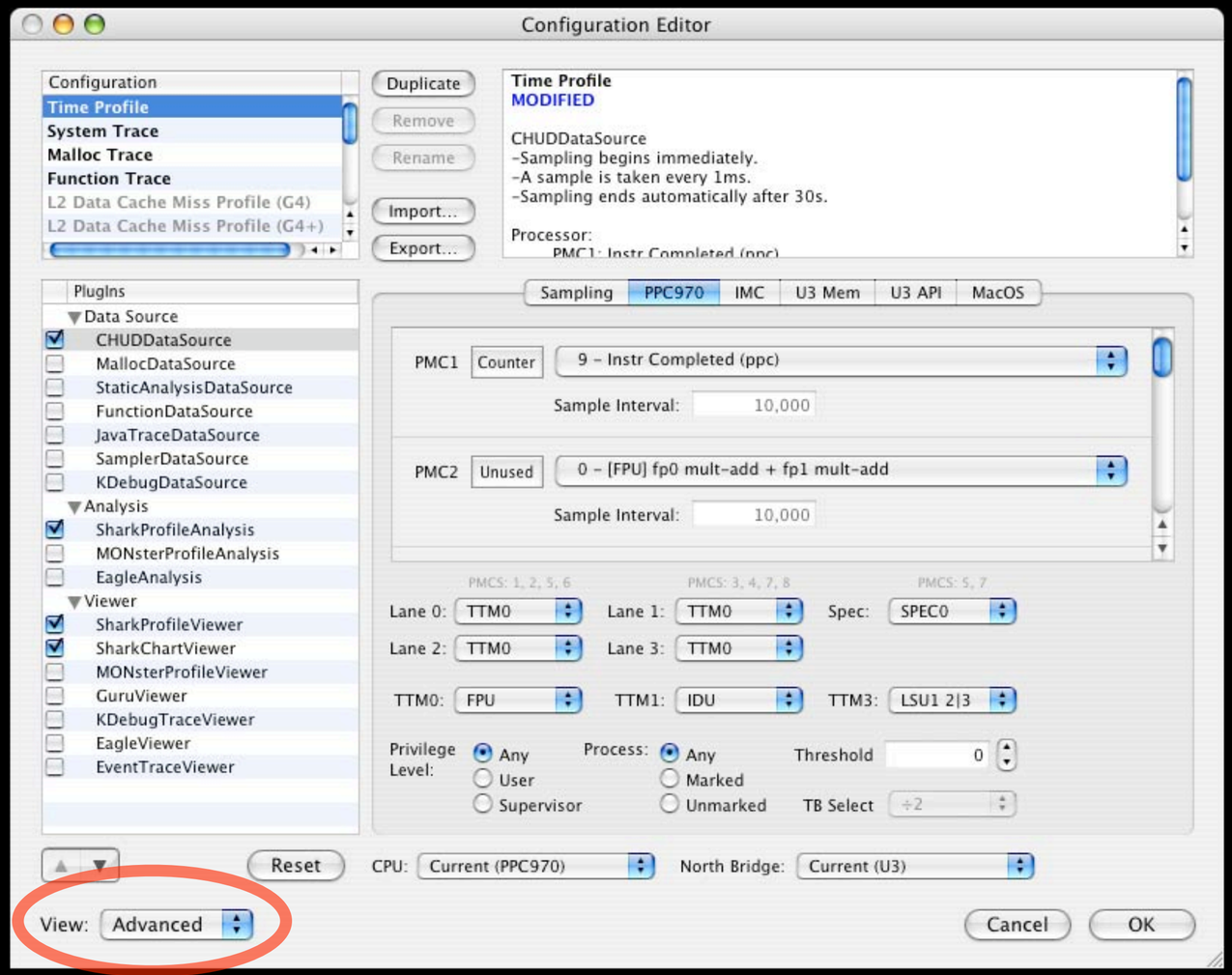
- Create your own configs
- Search event lists by name



Configuration Editor

The buck stops here

- Create your own configs
- Search event lists by name
- Control every aspect of profiling!



Getting Started

http://developer.apple.com/wwdc2005/

Performance Analysis 101

Step 1:

Establish a Baseline

Step 2:

Profile Optimized Deployment Code
Debug Symbols Enabled

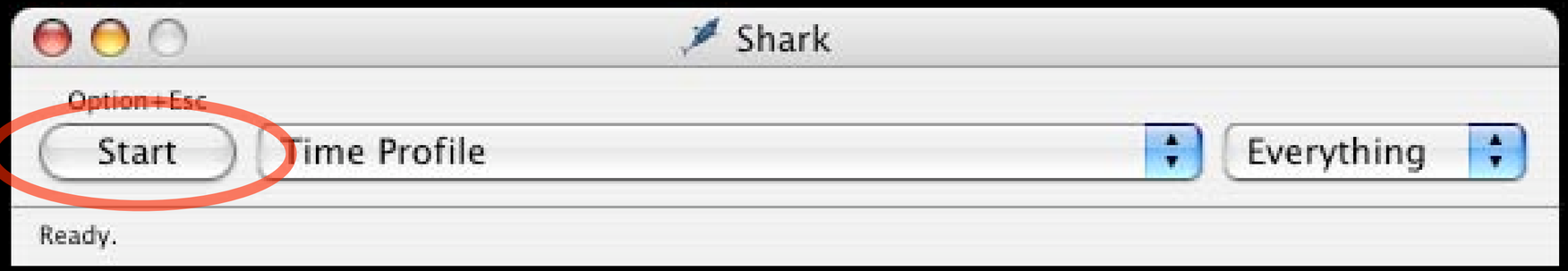
Step 3:

Shark It!

Getting Started With Shark

Just one click

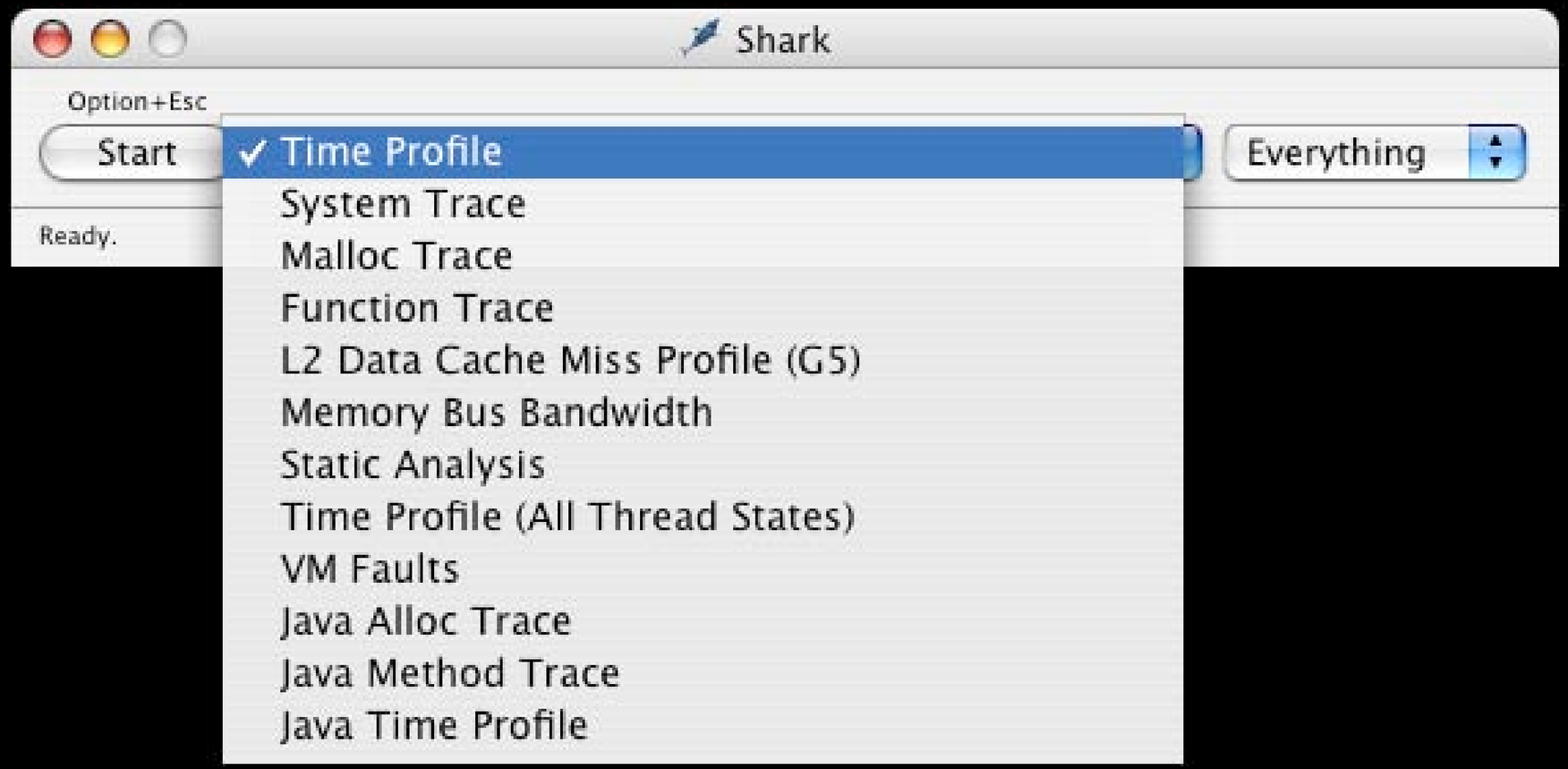
- Use **Start** button or hot key to begin profiling
- You're on your way…



Getting Started With Shark

Choose your weapon

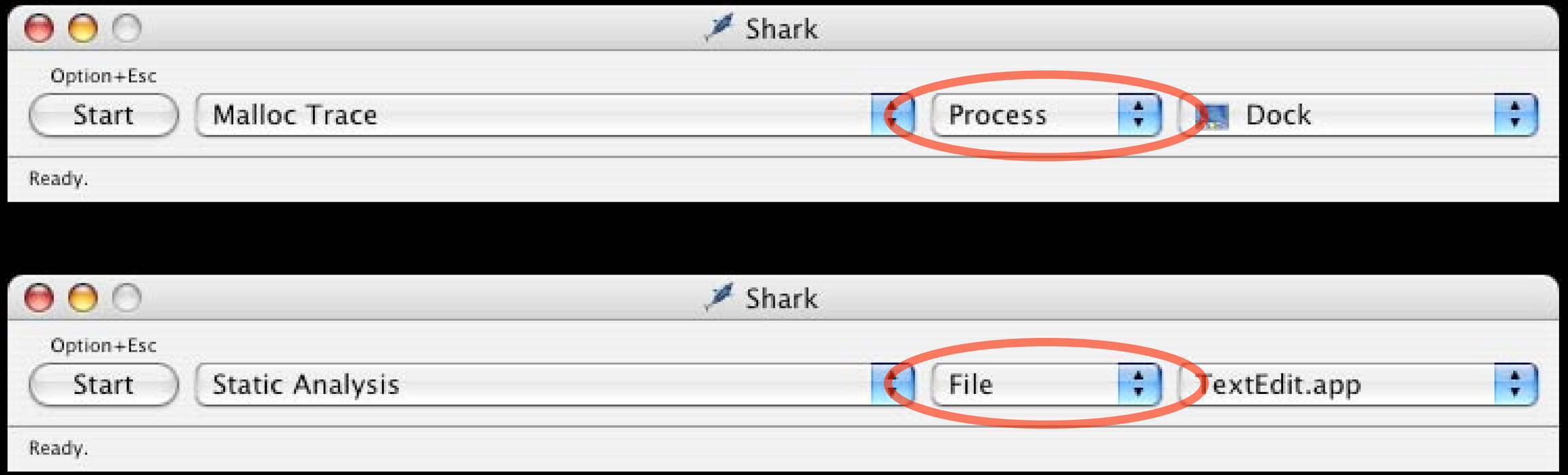
- Sampling configurations



Getting Started With Shark

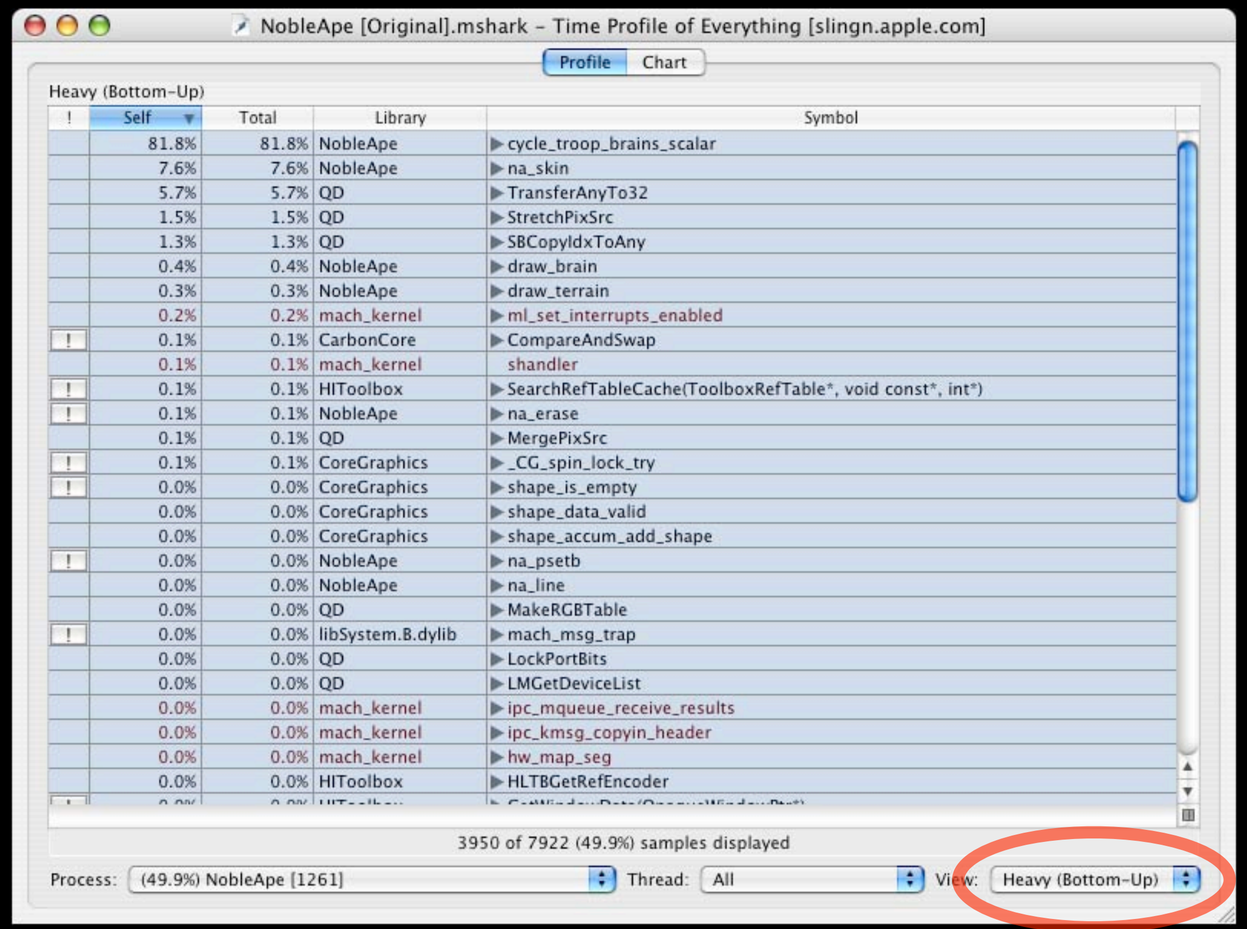
Choose your weapon

- Target everything, process, or file



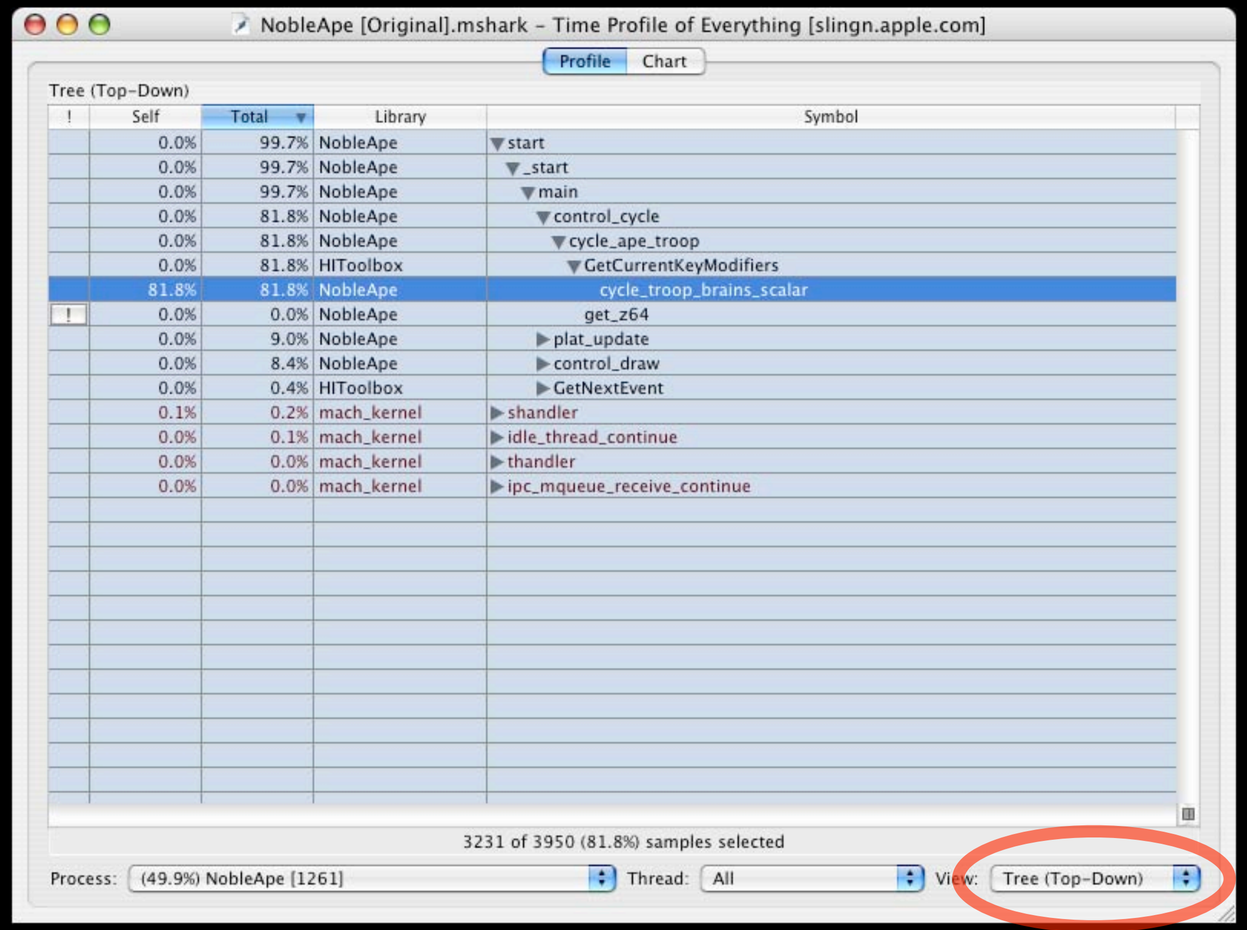
Profile Browser—Heavy View

The bottom line



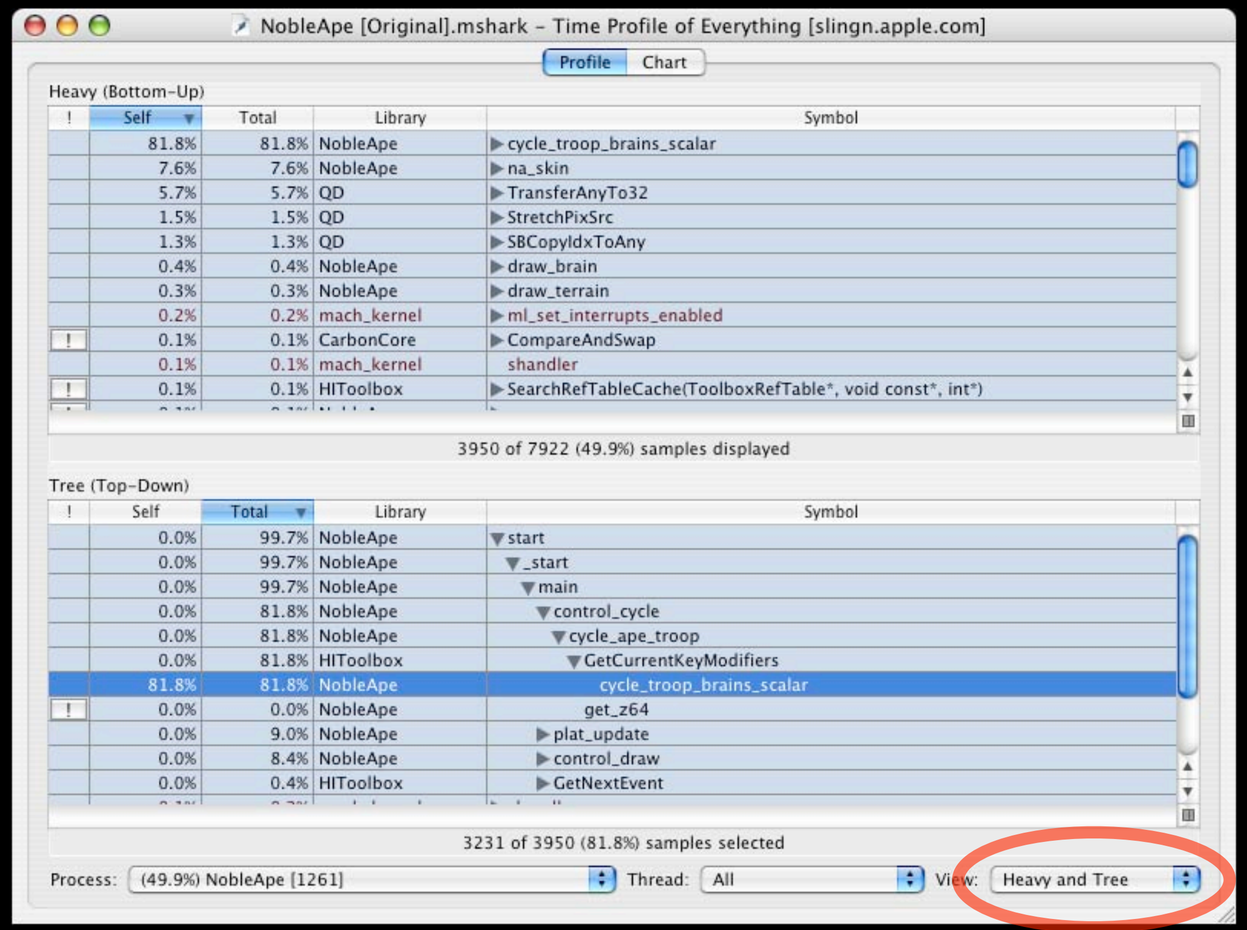
Profile Browser—Tree View

Getting there is half the fun



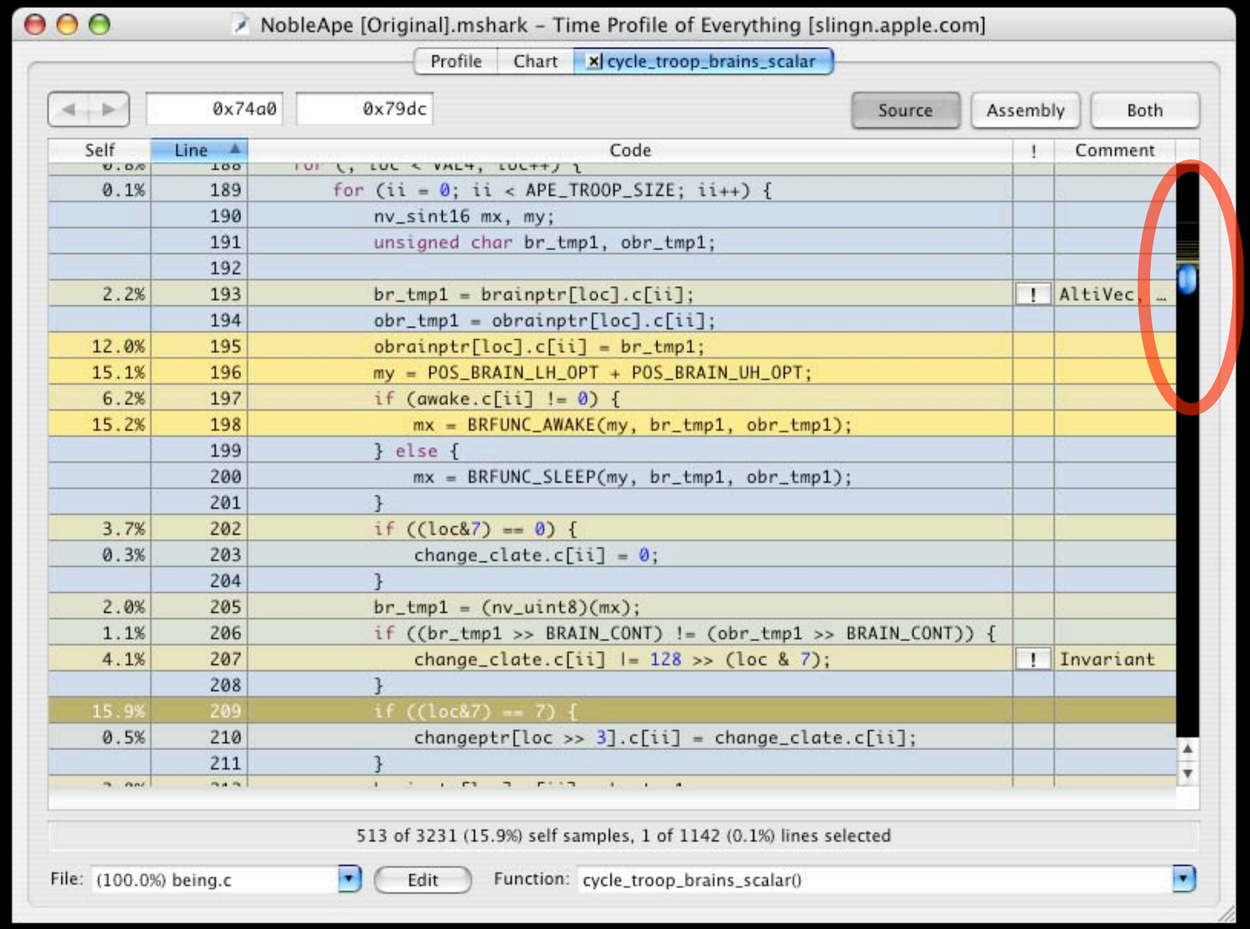
Profile Browser—Heavy and Tree

Where am I and how did I get here?



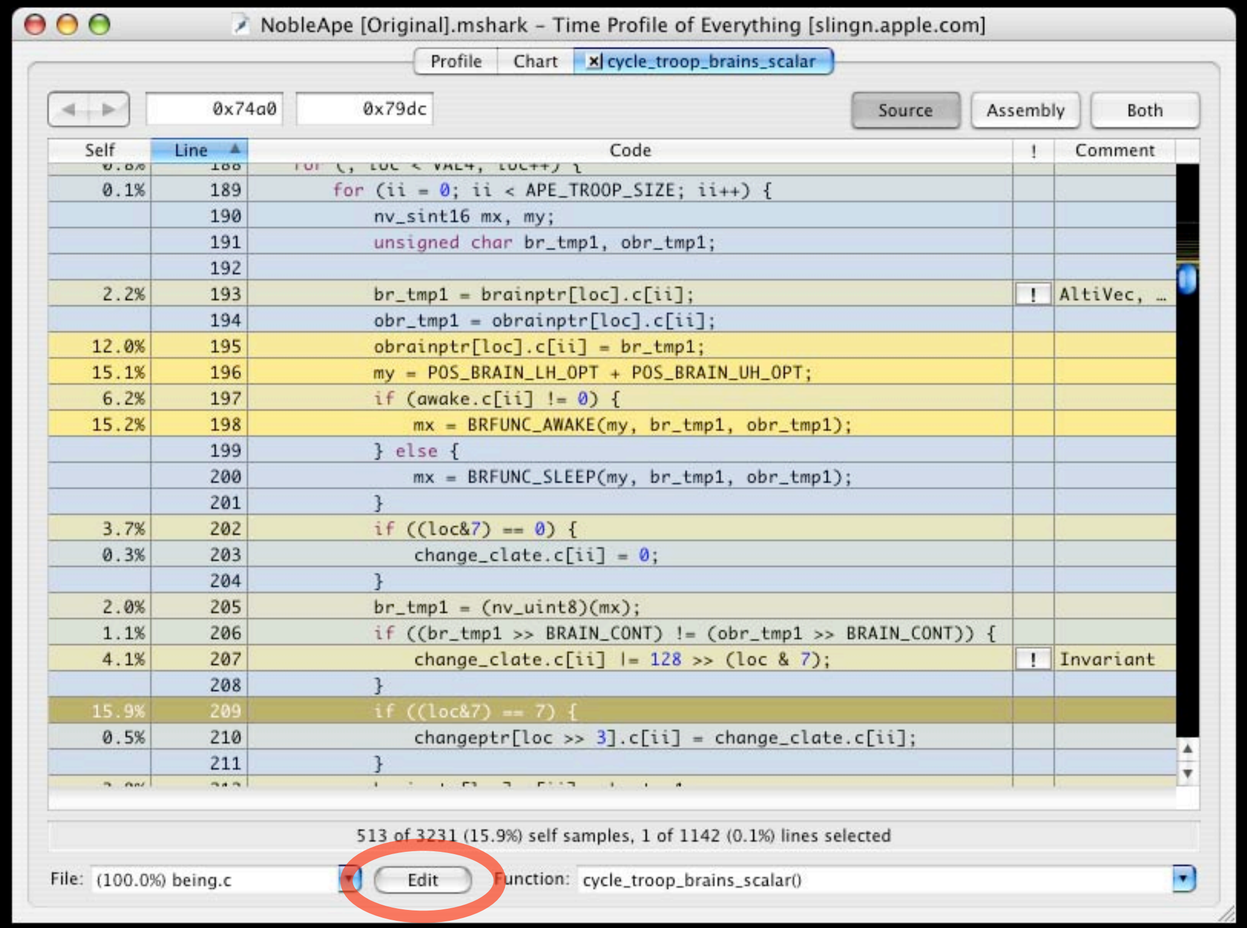
Code Browser

Here comes the Sun



Code Browser

Here comes the Sun



Profiling Networked Systems

http://developer.apple.com/wwdc2005/

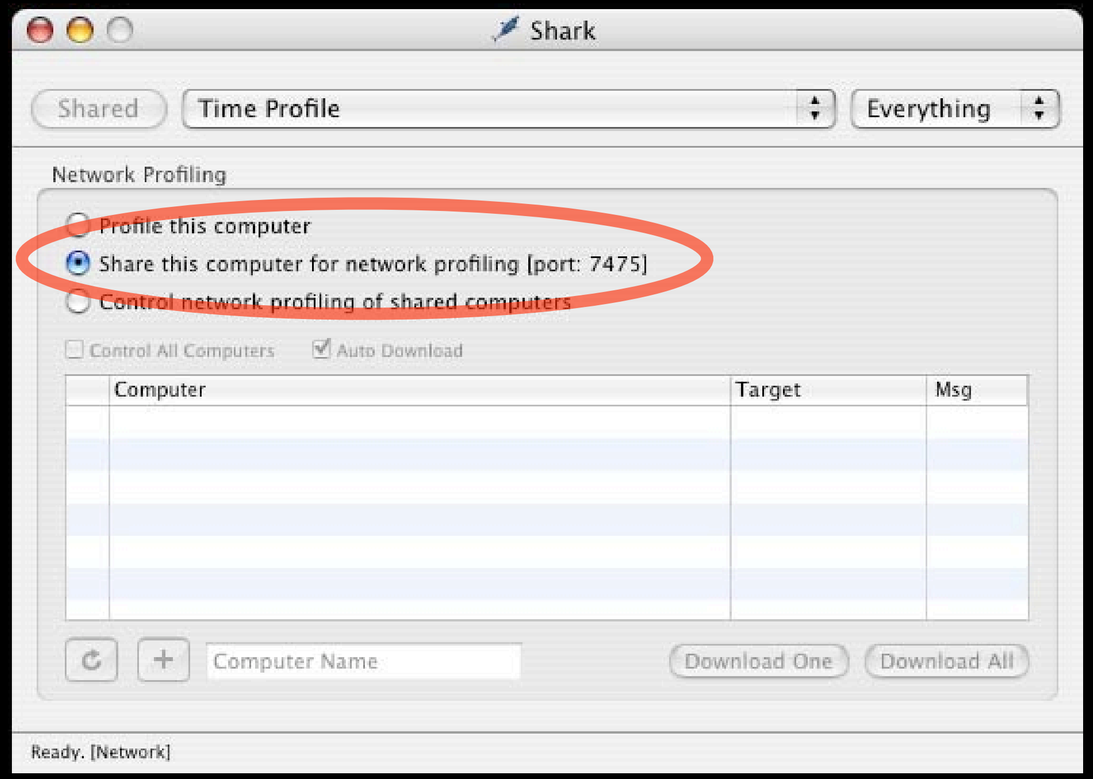
Profiling of Networked Systems

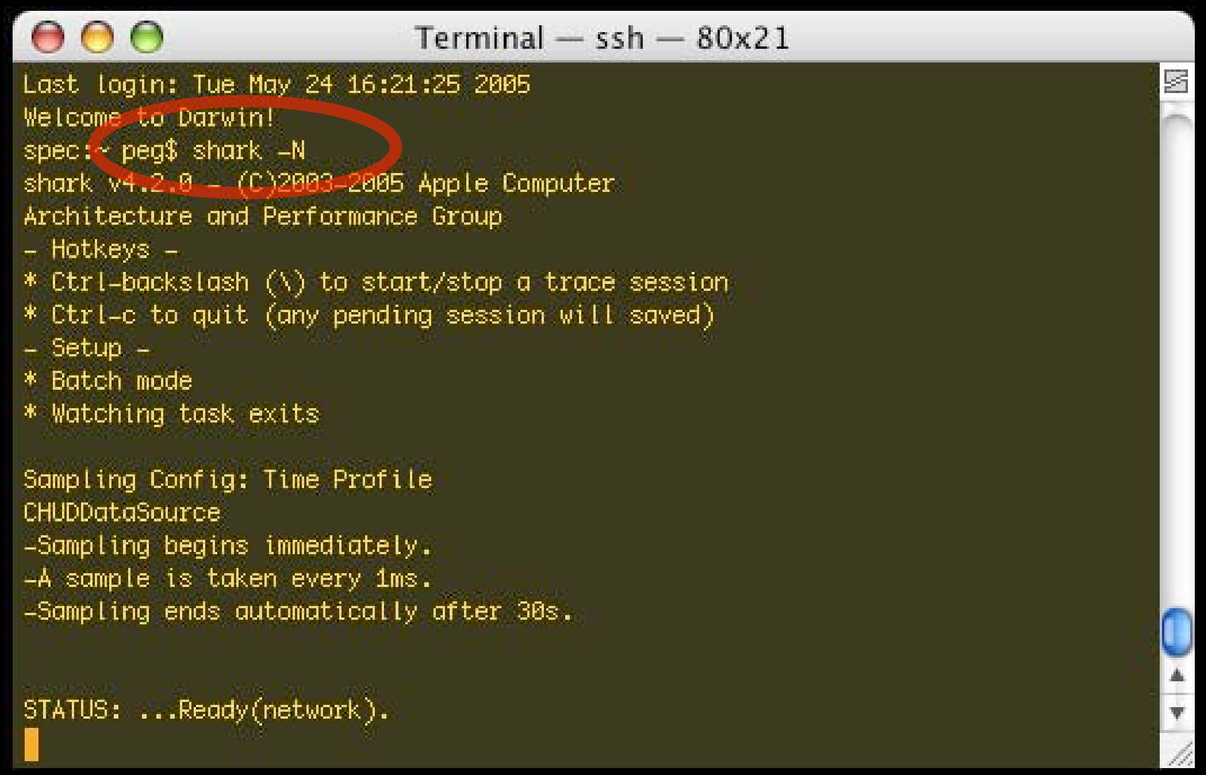
Your camera on the cluster

- Share computer for network profiling using
  - Shark Application
  - Command line Shark (/usr/bin/shark)
- Discover and control shared computers using
  - Bonjour (mDNS)
  - Static IP address lookup
- Simultaneously profile multiple machines
- Retrieve profile sessions automatically or on-demand

Profiling of Networked Systems

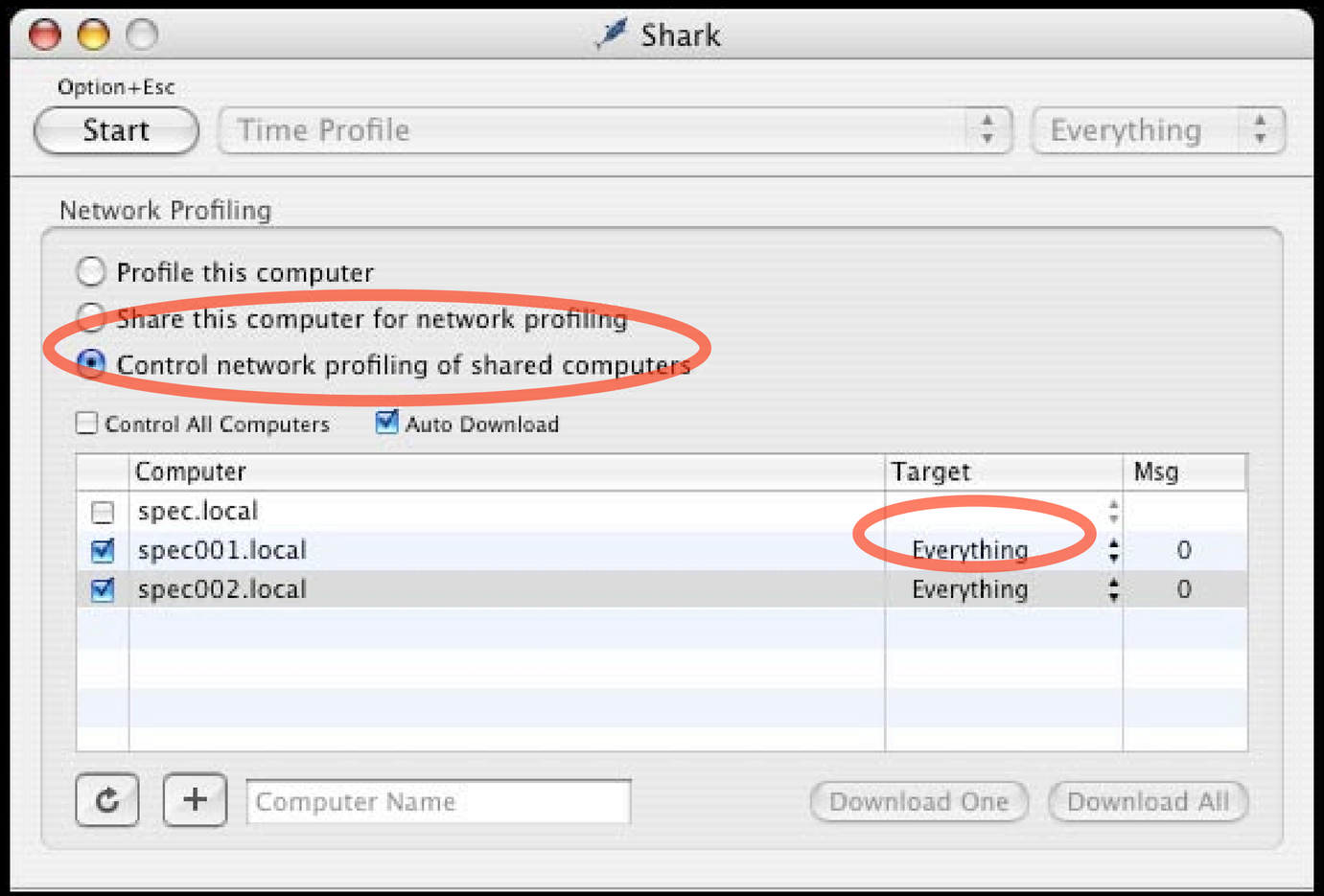
Sharing your computer for network profiling





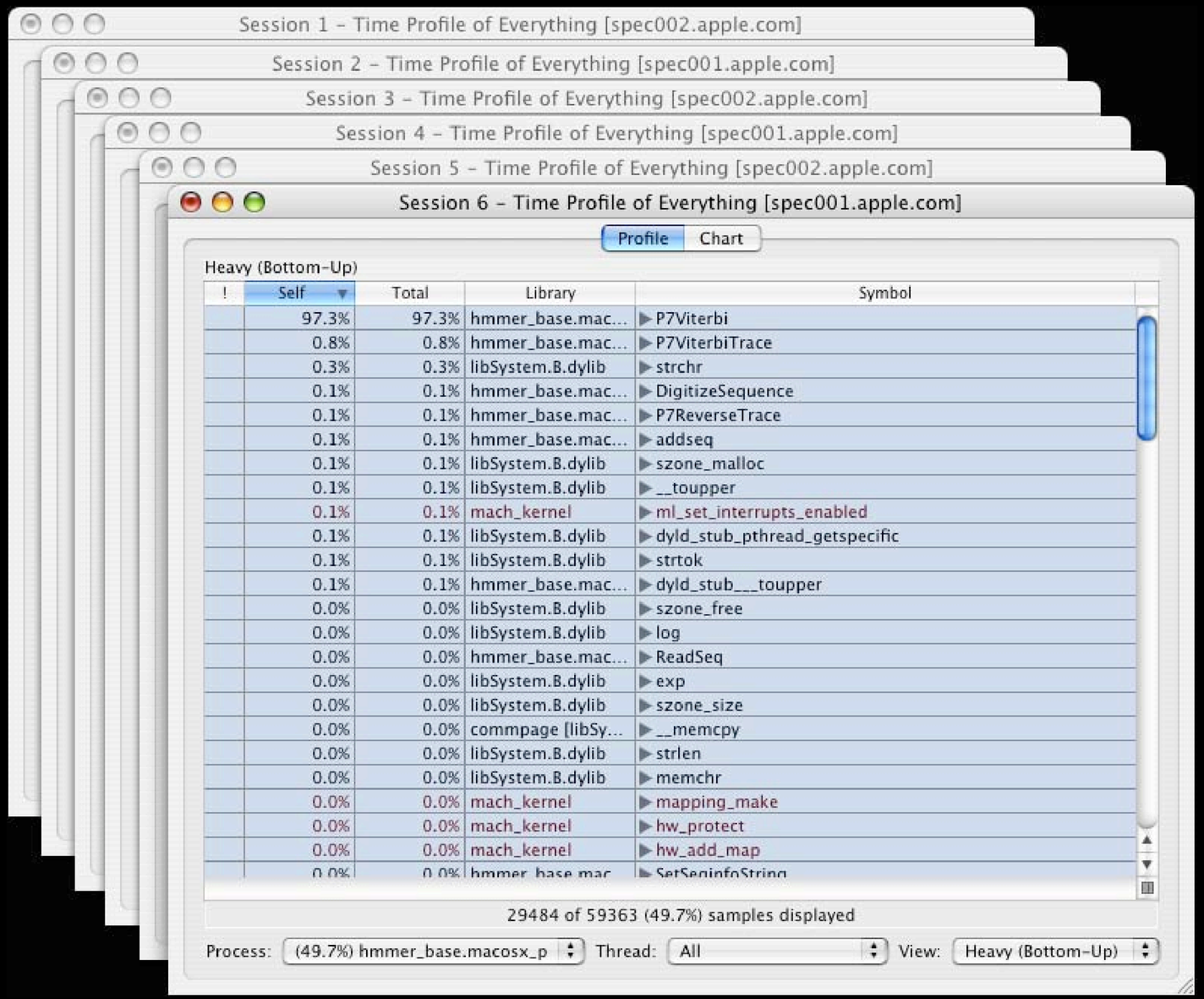
Profiling of Networked Systems

Discover and control shared computers Network profiling using Rendezvous



Profiling of Networked Systems

Profile multiple machines and retrieve sessions



Summary

Swimming with Shark

- Easy for beginners, powerful enough for experts
- Great for high-level and low-level performance analysis
- Compatible with all major Mac OS X compilers
- Available for free! (http://developer.apple.com/tools/performance/)
- Available today!

Absoft FxP Technology Preview

Rodney Mach
Absoft HPC Technical Director
rwm@absoft.com

http://developer.apple.com/wwdc2005/

FxP MPI Debugger Overview

- Basic entry level graphical MPI debugger
- Supports debugging 64-bit and 32-bit codes, both MPI and serial codes
- Familiar Fx2 serial debugger interface
- Support for Fortran/C/C++ and Assembler

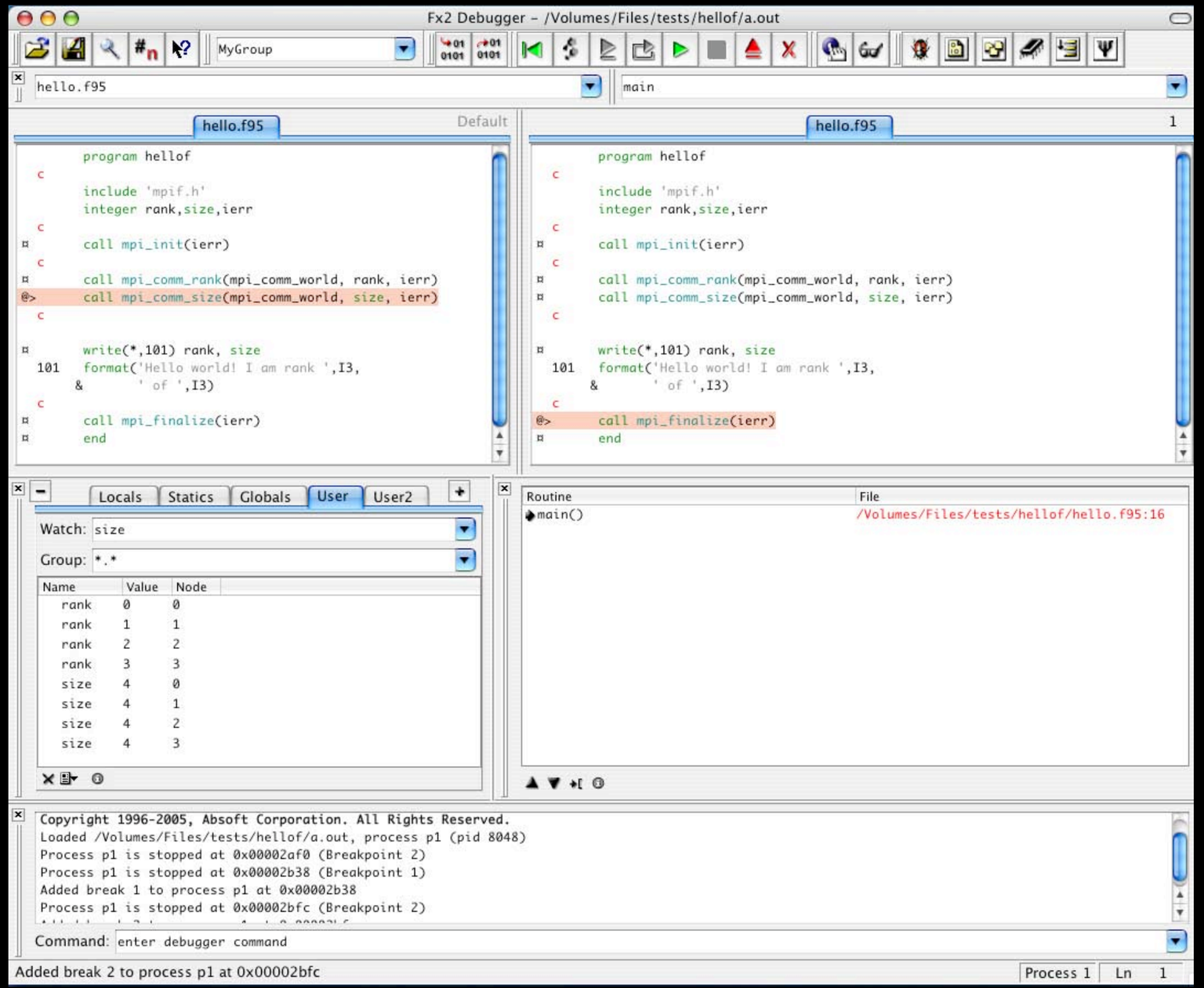
FxP MPI Debugger Benefits

- Supports all major compiler vendors and MPI implementations in one package
- Easy to use graphical user interface is simple to learn and highly intuitive for immediate productivity and reduced training costs
- Builds on market leading Fx2 serial debugger for Fortran, C/C++, and assembler

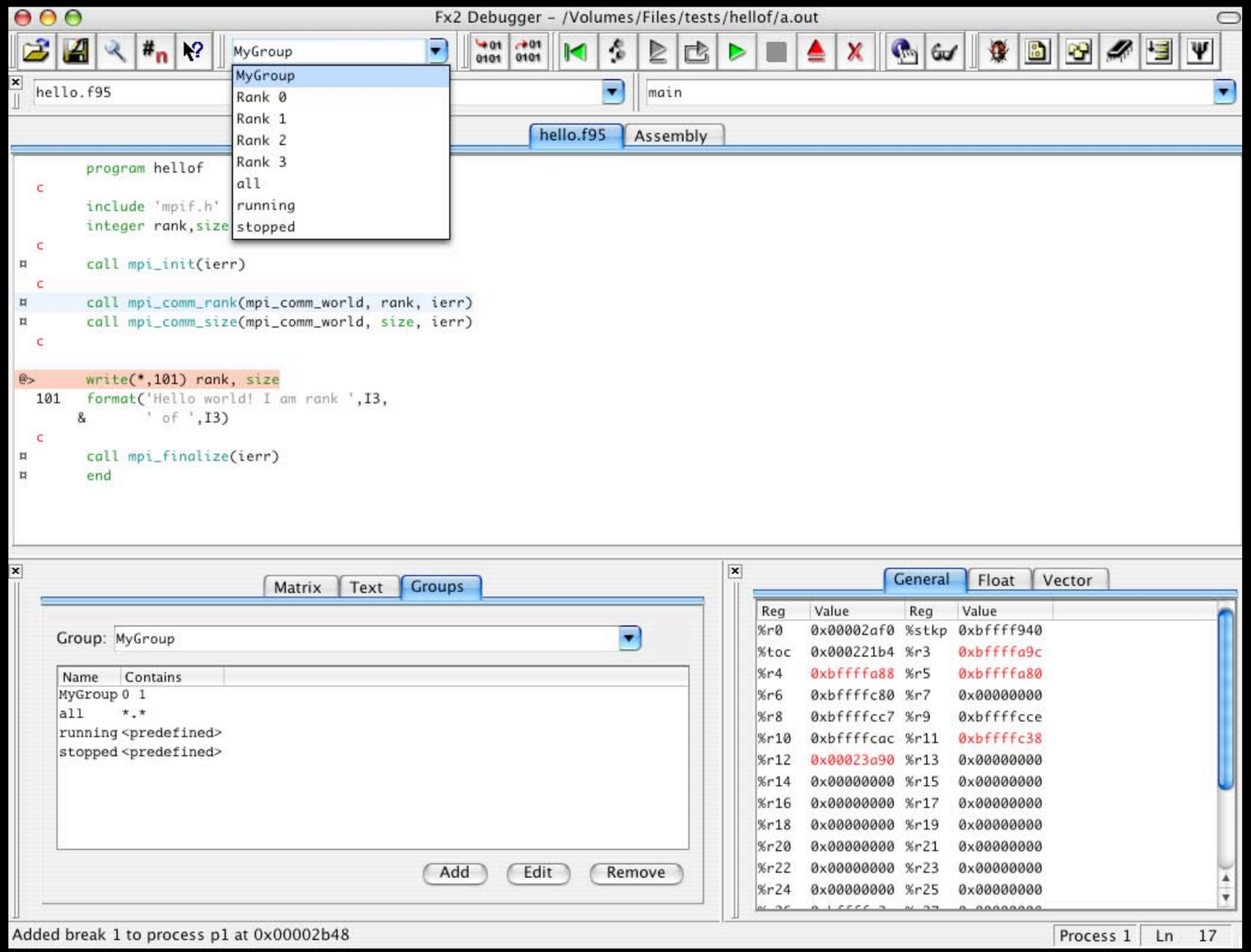
FxP MPI Debugger Features

- Basic MPI debugging support
- Automatically attach to MPI processes
- View local and global variables, stack trace, registers, message queues, across all MPI processes
- Visually see the state of all running processes
- The more powerful and fully featured parallel debugger TotalView also is available

FxP Main Window



FxP Named Groups



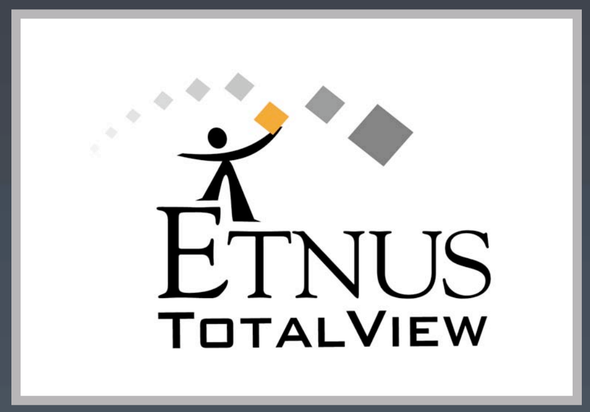
Absoft—Next Steps

- For sales inquiries, email sales@absoft.com
- For information, visit http://www.absoft.com
- For telephone inquiries, call:

Wood Lotz
(248) 853-0050

Etnus TotalView on Mac OS X

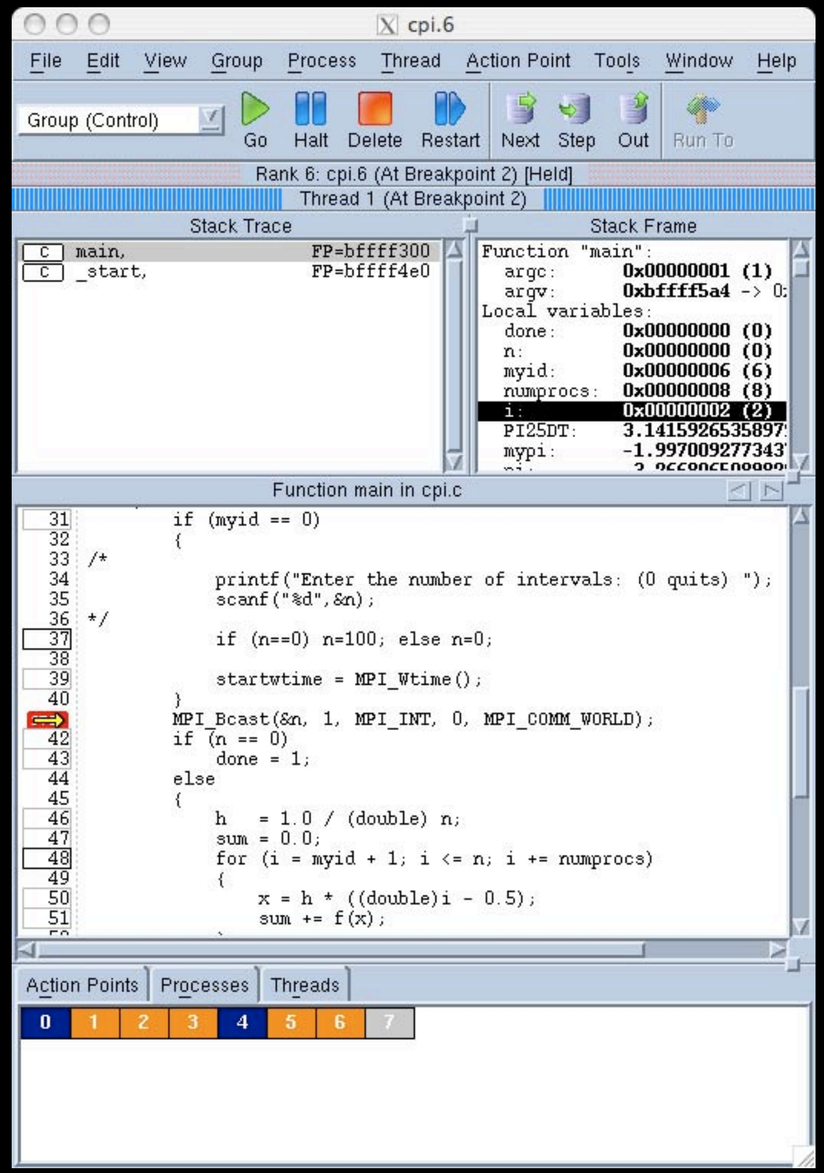
Chris Gottbrath
Partner Technologies Engineer
Etnus



http://developer.apple.com/wwdc2005/

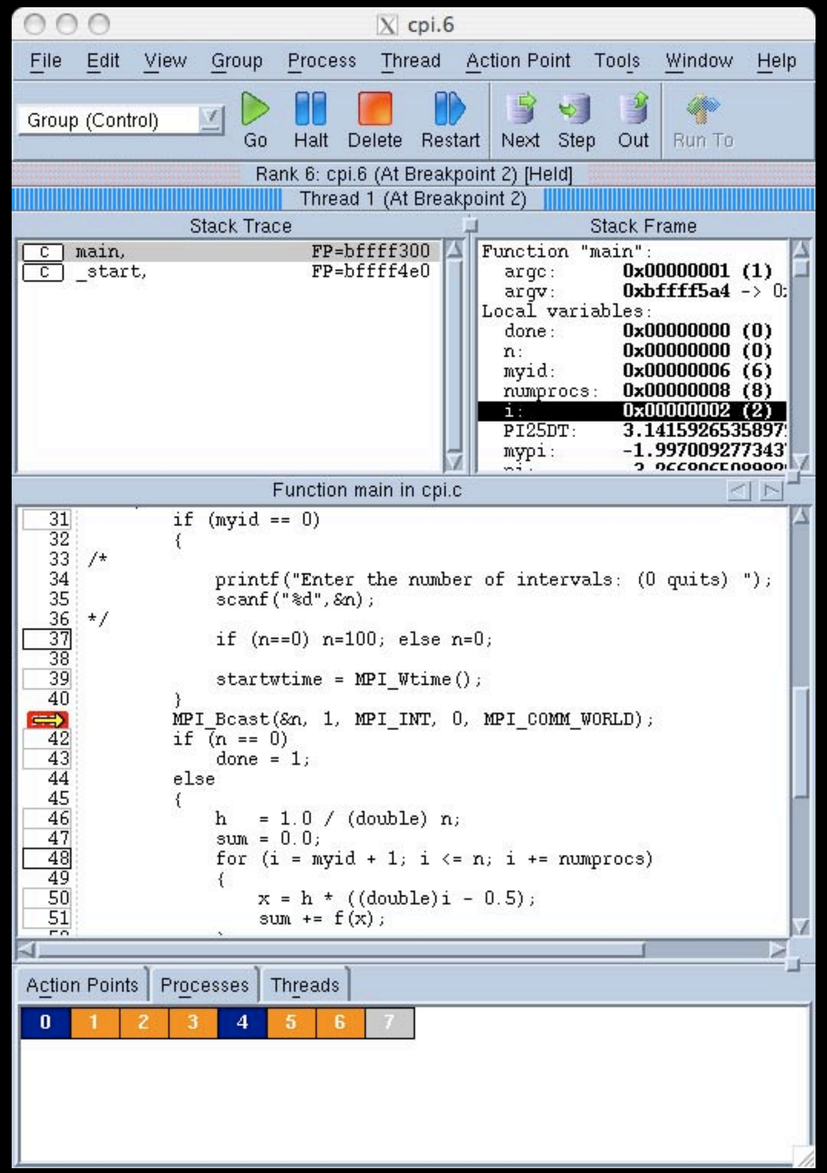
What Is Etnus TotalView?

- A major UNIX application dev tool
- Source Code Debugger
  - Serial or parallel
- C, C++, Fortran, Fortran90
  - Complex language features
- Wide compiler and platform support
- Multithreaded debugging, including OpenMP
- Distributed debugging
  - Cluster architecture
- X11 GUI and CLI



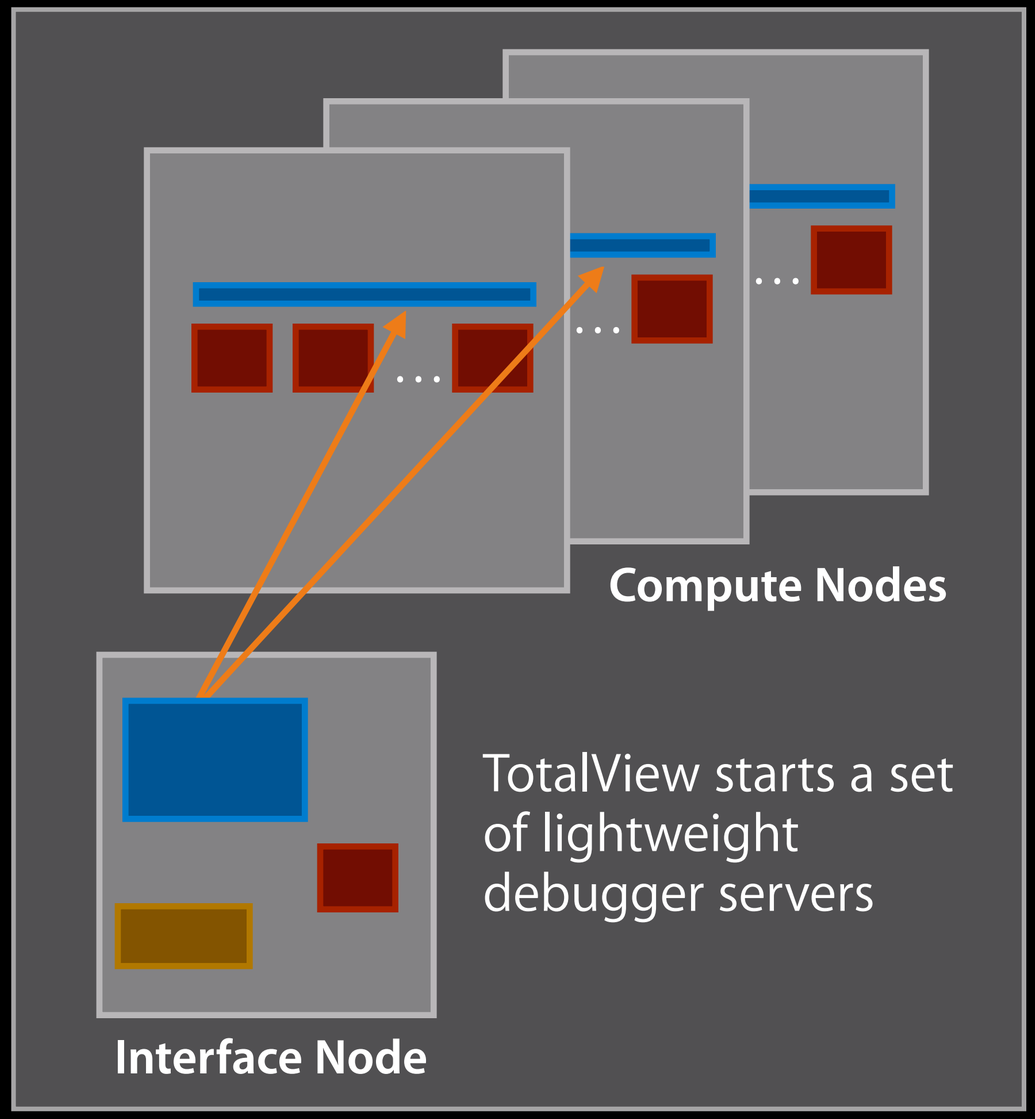
Etnus TotalView

- TotalView as a Parallel Debugger
- TotalView's Architecture
- Live Demo
  - Automatic Process Acquisition
  - Parallel Debugging Features
    - MPI Message Queue Debugging
- Scalability
- Roadmap



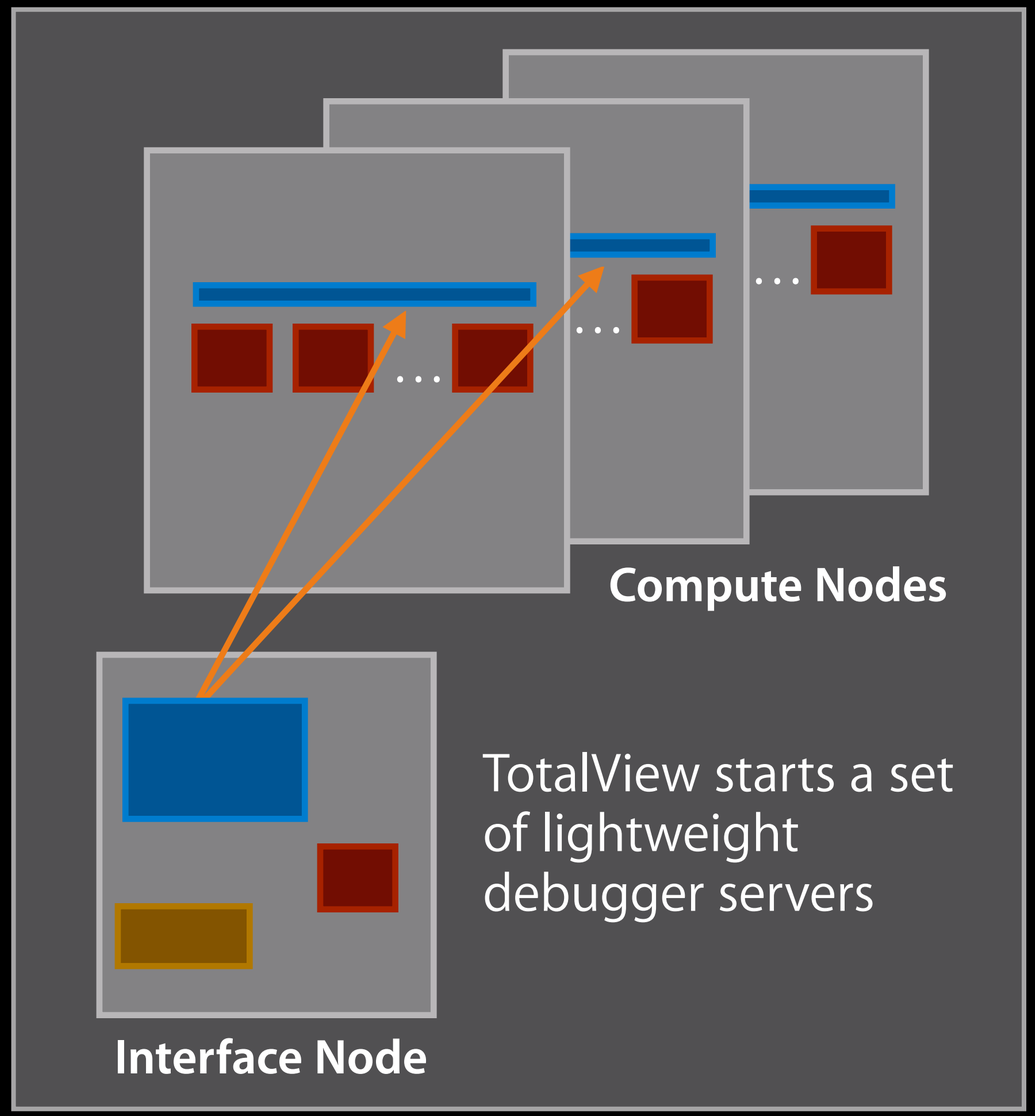
Architecture

- Single Client (TotalView)
  - Implements most of the functionality
  - GUI/CLI and debug engine
- Debugger Servers (tvdsvr)
  - One per node
  - Traces one or more rank processes
- TotalView starts and communicates directly with tvdsvrs



Architecture Implications

- Widely applicable model
- Low overhead on each node
- Does not use MPI for communication
- Debugger processes run as user
- Provides: Robust, Scalable, Minimal Interaction

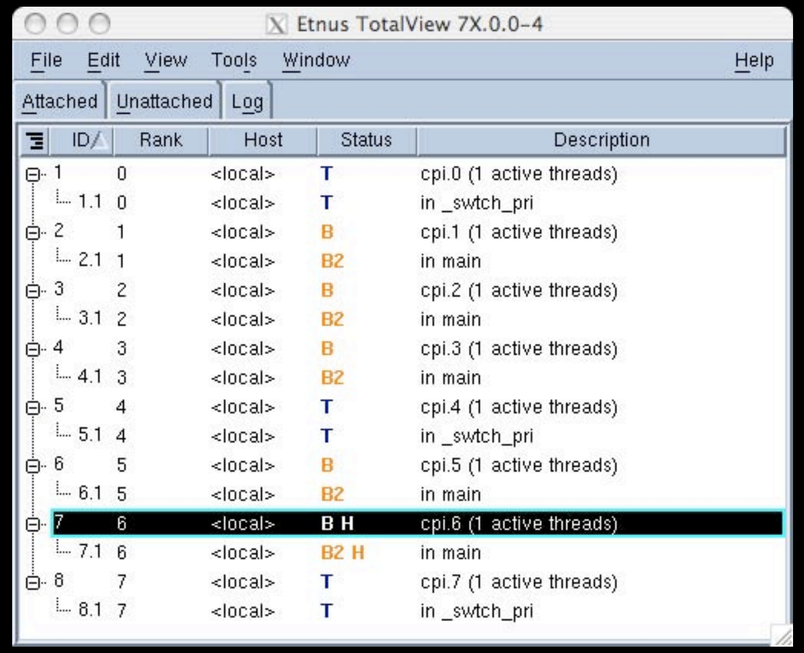


TotalView Demo

- TotalView 7 Beta running on a G4 powerbook
- Compiled with the Apple build of GCC 4
- Using a standard MPICH 1.2.6 Message Passing Library

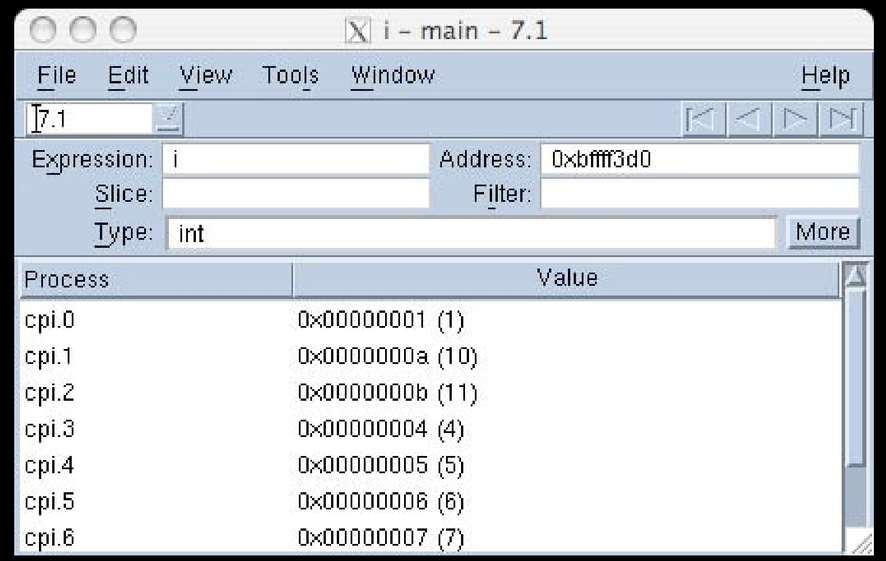
Process Acquisition

- Seamlessly attach to all the processes in an MPI job
- Jobs launched by TotalView, and already running or hung
- Public interface supported by most MPIs
- Subset attach
  - A scalability feature
  - Debug a subset of the MPI processes
  - Interactively attach and detach from processes
  - Use a “fan out” strategy



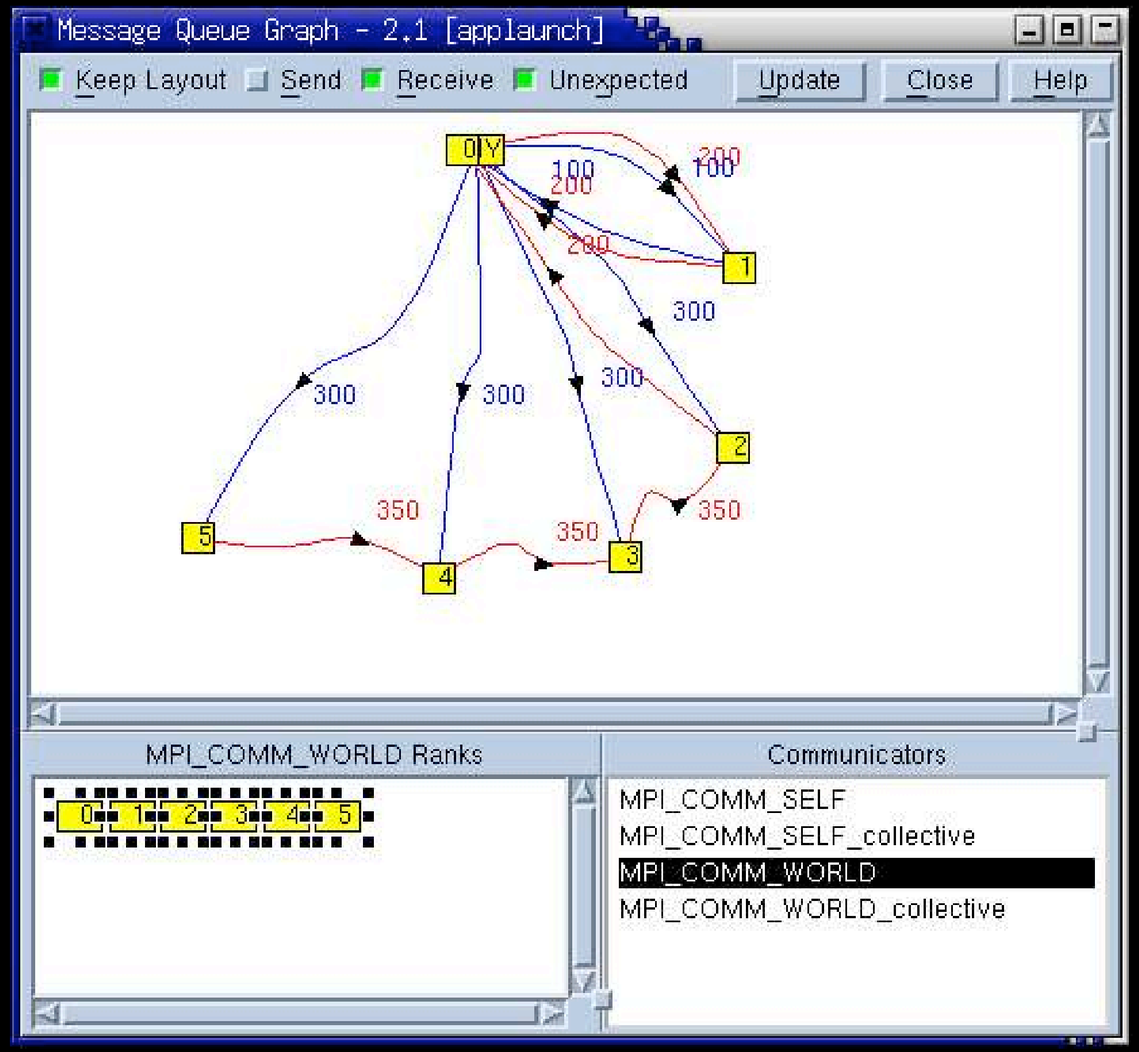
Parallel Debugging Features

- Easy to see and control program state
  - Work with individual process or groups of processes
- View SIMD data across all processes from one window
- Parallel expression evaluation
- Rich set of action points
  - Per thread, process or group granularity
  - Barrier breakpoints
  - Evaluate C, C++, Fortran code fragments



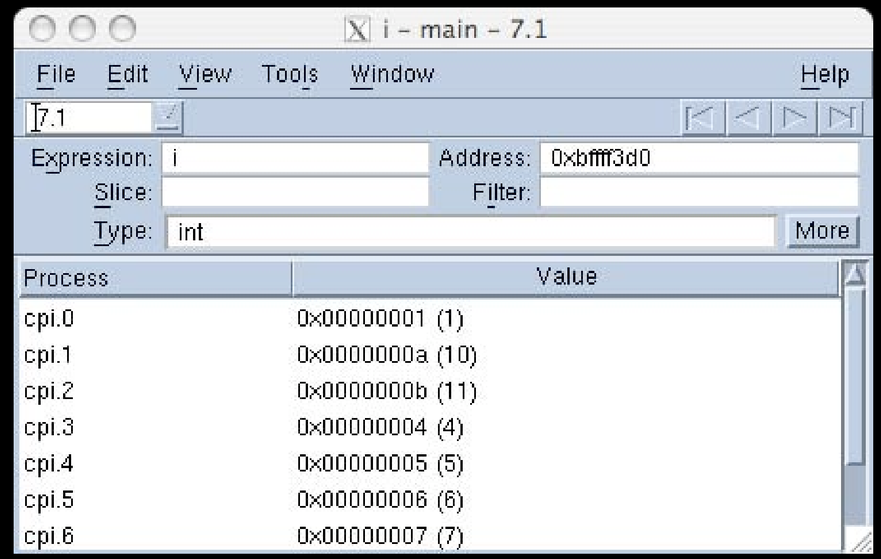
MPI Message Queue Information

- Shows MPI communications state
- MPI programs can suffer deadlocks
- State hidden in MPI library
  - Needed to understand
- Public interface that many MPI vendors support
- Message Queue graph
- Message Queue detail windows



Scalability

- Scalability means many things
  - Performance/responsiveness
  - Memory usage
  - Status and data representation
  - Control Issues
  - Program size/complexity also grows
- Practical scalability
  - 10s to 100s of processes trivially
  - 1,000s of processes can be debugged currently with TotalView
  - Scalability is an ongoing concern



Etnus TotalView 2005 Releases

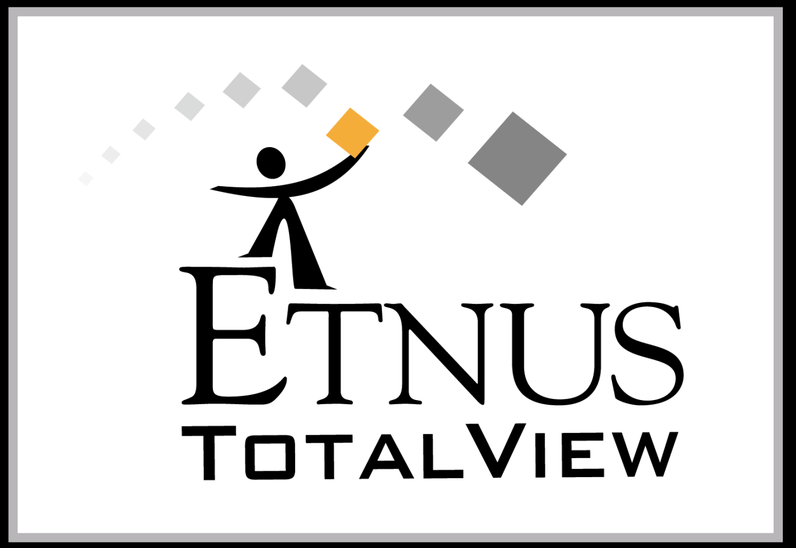
- Two major feature releases in 2005
- Four minor point releases throughout 2005
- TotalView 7.0
  - C++ and F90 Expression system
  - MPI usability update
  - Introduces Apple OS X on power-darwin support!
  - Currently in Beta
- TotalView 7.x
  - More C++ features: exception handling, dynamic class hierarchy casting
  - MPI, UI, and memory debugging improvements
  - Q4 2005

Etnus TotalView on Mac OS X Features

- In TV 7
  - Final stage of beta release cycle now
- Supports both Tiger and Panther
- Supports X Code by virtue of supporting
  - Apple GCC builds, Absoft, and IBM XL compilers
- Supports X Grid by virtue of supporting
  - MPICH and LAM MPI
- Heap Memory Debugging feasibility analysis underway
- Would like to begin looking at Intel-darwin

To Contact Etnus

- See www.etnus.com
  - Sign-up for a webcast
  - Sign-up for our tip-of-the-week emails
  - Get a free, fully-functional trial version
- Contact sales@etnus.com
  - For pricing and configuration information
- Contact support@etnus.com
  - For technical issues
  - To let us know what you like or don't like



More Information

http://developer.apple.com/wwdc2005/

Documentation, Sample Code, and Other Resources

http://developer.apple.com/wwdc2005/

WWDC2005

Worldwide Developers Conference